UNIVERSITÉ DE MONTRÉAL

CONTRIBUTIONS À L'ANALYSE STRATÉGIQUE DU TRANSPORT DE MARCHANDISES : CALCUL DU FLOT MAXIMAL MULTIPRODUIT ET MODÉLISATION PORTUAIRE

par

Nathalie Rico

CENTRE DE RECHERCHE SUR LES TRANSPORTS

MÉMOIRE D'ÉTUDIANT

CASE POSTALE 6128, SUCCURSALE A, MONTRÉAL, QUÉBEC H3C 3J7

CONTRIBUTIONS À L'ANALYSE STRATÉGIQUE DU TRANSPORT DE MARCHANDISES : CALCUL DU FLOT MAXIMAL MULTIPRODUIT ET MODÉLISATION PORTUAIRE

par

Nathalie Rico

Rapport de stage de maîtrise, département d'informatique et recherche opérationnelle, dirigé par Michael Florian.

Cette étude a été publiée grâce à une subvention du fonds F.C.A.R. pour l'aide et le soutien à la recherche.

Université de Montréal

Centre de recherche sur les transports - Publication #509

Juin 1986

Dépôt légal - 1er trimestre 1987

SOMMAIRE

La planification stratégique du transport de marchandises à une échelle nationale consiste à faire une analyse de la situation présente et d'évaluer les impacts de modifications futures sur l'ensemble du réseau de transport. L'outil informatique pour aider à la planification comprend les aspects suivants : il permet de traiter des réseaux à plusieurs modes et à plusieurs produits ainsi que de simuler les flots de marchandises soit un mode de transport particulier, soit sur une combinaison quelconque de modes. Pour ce faire, il est nécessaire de connaître la quantité maximale de produits pouvant être envoyée des origines aux destinations à travers le réseau à capacité limitée. De plus, étant donné que cette congestion découle en grosse partie des ports, il est important de modéliser les ports pour reproduire toutes les activités qui s'y déroulent.

La première partie de ce rapport traite par conséquent du calcul du flot maximal multiproduit dans un réseau à capacité limitée. Le problème est énoncé et la méthode de résolution est explicitée. La méthode est une heuristique basée sur la relaxation des contraintes de capacité : la méthode du lagrangien augmenté. Le sous-problème avec coûts convexes est résolu par une approche Frank & Wolfe. Des extensions du modèle de base sont également exposées et des résultats sont présentés sur des réseaux de petite taille.

La deuxième partie de ce rapport concerne la modélisation des ports. Une étude des mouvements et des activités se déroulant dans les ports a été effectuée dans le but de trouver une représentation des ports et une méthodologie pour effectuer des analyses adéquates.

Ce rapport s'inscrit dans le cadre d'un projet de collaboration avec le Brésil et a pour but de développer un outil informatique pour aider à la planification stratégique du transport de marchandises.

TABLE DES MATIERES

page

Sommaire....ii
Table des matières....iv
Liste des figures....vii
Liste des tableaux....viii

INTRODUCTION....1

Chapitre 1 : CALCUL DU FLOT MAXIMAL MULTIPRODUIT....4

1.1 Enoncé du problème....4
1.2 Revue de littérature....6
1.2.1 Décomposition basée sur l'allocation des prix....8
1.2.2 Décomposition basée sur l'allocation des ressources....10
1.2.3 Approche de partitionnement....16
1.2.4 Méthode du lagrangien augmenté....17
1.3 Approche lagrangienne augmentée....17
1.3.1 Forme générale du lagrangien augmenté....18
1.3.1.1 Contraintes d'égalité....18
1.3.1.2 Contraintes d'inégalité....21
1.3.2 Lagrangien augmenté appliqué au problème du flot maximal multiproduit....22
1.4 Technique de résolution....27
1.4.1 Conditions Kuhn-Tucker....27
1.4.2 Algorithme de type PARTAN....28
1.4.3 Détails d'implantation....32
1.5 Extensions du modèle....33
1.5.1 Flot fixé sur chaque lien....33
1.5.1.1 Enoncé du problème....33
1.5.1.2 Modifications de l'algorithme....34
1.5.2 Capacités dépendant du flot dans les deux directions....35
1.5.2.1 Enoncé du problème....35
1.5.2.2 Modifications de l'algorithme....37
1.5.3 Capacités différentes par combinaison de produit....39
1.5.3.1 Représentation du réseau....39
1.5.3.2 Produits permis sur un lien....40
1.5.3.3 Modifications de l'algorithme....42

1.5.4 Borne inférieure sur la demande de certaines paires O/D pour certains produits....44
1.5.4.1 Enoncé du problème....44
1.5.4.2 Première méthode de résolution : approche lagrangienne augmentée....46
1.5.4.3 Deuxième méthode de résolution : modification directe de l'algorithme de résolution....51
1.6 Résultats numériques....55
1.6.1 Réseau Dupuis....55
1.6.2 Choix des paramètres....55
1.6.3 Variation du facteur de pénalité PEN....58
1.6.4 Variation du poids γ_i^p de chaque paire O/D et de chaque produit....59
1.6.5 Variation de la demande maximale $g_i^{p\,max}$....61
1.6.6 Flot fixé sur chaque lien....65
1.6.7 Capacités dépendant du flot dans les deux directions....66
1.6.8 Capacités différentes par combinaison de produit....69
1.6.9 Borne inférieure sur la demande de certaines paires O/D pour certains produits : comparaison entre les deux méthodes de résolution....69

Chapitre 2 : MODELISATION PORTUAIRE

2.1 Motivation....73
2.2 Analyse et Modélisation d'un port....74
2.3 Mouvements de marchandises dans un port....75
2.4 Simplifications apportées au modèle....77
2.4.1 Première simplification....77
2.4.2 Deuxième simplification....80
2.4.1 Représentation du réseau portuaire....80
2.4.2 Mouvements de la marchandise....81
2.4.3 Inter-relation entre les ports....86
2.4.4 Fonctions de transfert....86
2.4.5 Fonctions de délai basées sur le volume....90
2.5 Entreposages....93
2.5.1 Entreposage dans le véhicule de transport....93
2.5.2 Entreposage dans le port à l'intérieur d'une même période d'analyse....93
2.5.3 Entreposage dans le port entre deux périodes d'analyse consécutives....94
2.6 Conclusions....97

CONCLUSION 99

Bibliographie
Remerciements

Annexe A : Algorithme PARTAN
Annexe B : "Listing" du programme MAXFLO

LISTE DES FIGURES

page

Figure 1.1 : Ancienne et nouvelle représentation du réseau pour considérer les capacités différentes par combinaisons de produits.....41

Figure 1.2 : Réseau "DUPUIS".....56

Figure 1.3 : Réseau test pour des capacités dépendant du flot dans les deux directions.....68

Figure 1.4 : Réseau test pour des capacités différentes par combinaisons de produits.....70

Figure 2.1 : Mouvements de la marchandise dans un port.....78

Figure 2.2 : Représentation générale d'un port : importation et exportation.....79

Figure 2.3 : Etapes de simplification du réseau portuaire.....82

Figure 2.4 : Représentation simplifiée d'un port.....83

Figure 2.5 : Représentation de la connection de trois ports.....87

Figure 2.6 : Représentation de la connection de deux ports.....88

Figure 2.7 : Fonction de coût pour modéliser la capacité.....91

LISTE DES TABLEAUX

page

Tableau 1.1 : Tableau de correspondance entre la variable IPROD et les produits permis sur le lien....43

Tableau 1.2 : Variation du facteur de pénalité PEN....60

Tableau 1.3 : Variation du poids des paires O/D i et des produits p....62

Tableau 1.4 : Variation des demandes maximales $g_i^{p\,max}$....64

Tableau 1.5 : Flot fixé sur chaque lien....67

Tableau 1.6 : Capacités dépendant du flot dans les deux directions....68

Tableau 1.7 : Borne inférieure sur la demande de certaines paires O/D pour certains produits : comparaison entre les deux méthodes de résolution....72

INTRODUCTION

L'analyse et la planification stratégique du transport de marchandises à une échelle nationale comporte les aspects suivants : faire une étude de la situation actuelle, cerner les problèmes et les besoins, et évaluer les impacts de différentes politiques sur l'ensemble du réseau de transport. Pour aider et faciliter la réalisation de ces tâches, il est utile de pouvoir étudier et comparer différents scénarios qui correspondent à la réalisation de divers projets ou aux différentes politiques désirées. On peut par exemple analyser la rentabilité et l'efficacité de projets tels la construction de nouvelles installations, la modification des coûts dans le réseau ou l'augmentation de la capacité d'une partie de celui-ci. Ces études ont pour but l'amélioration globale du réseau de transport de marchandises.

Ce rapport de stage s'inscrit dans le cadre d'un projet du Centre de Recherche sur les Transports de l'Université de Montréal en collaboration avec l'Université catholique de Rio de Janeiro (P.U.C. - R.J.) et d'un organisme d'études en transport attaché au Ministère des Transports Brésiliens (GEIPOT). Une équipe au Centre de Recherche sur les Transports travaille à l'élaboration d'un outil informatique pour aider à la planification stratégique du transport de marchandises dans tout le Brésil. Ce rapport de stage vise essentiellement deux aspects de l'analyse du transport de marchandises : le calcul du flot maximal multiproduit et la modélisation portuaire.

Cet outil de planification est basé sur des méthodes développées en recherche opérationnelle et son utilisation sera facilitée grâce à une interface interactive et graphique avec l'usager. Il deviendra dès lors possible entre autres d'afficher une partie du réseau de transport, de simuler une situation donnée, de

comparer plusieurs scénarios etc... afin d'aider les planificateurs lors de la prise de décision.

Décrivons le réseau de transport considéré. A la base, nous considérons un réseau de transport G=(N,A) où N est l'ensemble des noeuds et A est l'ensemble des liens. Ce réseau permet plusieurs modes de transport dont les principaux sont le mode routier, le mode ferroviaire et le cabotage. Il permet également le traitement de plusieurs produits ou catégories de produits. Le réseau comprend des ports pour l'importation et l'exportation de la marchandise et pour le transport d'un port à un autre port du pays. Nous avons également la présence de corridors d'importation et d'exportation : ces corridors sont des portions du réseau où la marchandise circule pour accéder de l'intérieur du pays aux ports et vice versa. Ces concentrations de trafic engendrent une congestion dans la partie du réseau correspondant à ces corridors ainsi que dans les ports où la capacité est restreinte.

Quels sont les principes de l'outil de planification stratégique développé ?

Les modèles considérés dans notre outil informatique sont des modèles permettant de simuler les flots de marchandises soit sur un mode de transport particulier tels le mode routier ou le mode ferroviaire, soit sur tous les modes à la fois, ou soit sur un sous-ensemble de modes. La simulation se fait de façon statique et non dynamique, c'est-à-dire que l'analyse s'effectue sur une certaine période de temps déterminée préalablement. Le projet de modélisation du transport de marchandises comprend donc plusieurs aspects, dont : la simulation des flots de marchandise sur le réseau routier, sur le réseau ferroviaire, sur tous les modes combinés, ou sur une combinaison de modes de transport, une analyse de prévision des matrices Origine/Destination (O/D) ainsi qu'une étude de la distribution des wagons vides dans le réseau. Lors de la simulation des flots de marchandise, il est possible que l'on ne puisse pas affecter toute la demande de certaines paires O/D sur ce réseau, à cause de la congestion dans le réseau ferroviaire et de la capacité restreinte dans les ports. D'où l'intérêt de connaître une estimation du flot maximal de produits pouvant

être envoyé de chaque origine vers chaque destination. Cette estimation nous indiquera la quantité de marchandise n'ayant pas pu être envoyée et qui sera entreposée entre deux périodes de temps consécutives.

Les ports constituent une partie très importante du réseau à cause des très nombreuses importations et exportations de marchandises et de la congestion qui en découle. En effet, dans le cas particulier du transport de marchandises au Brésil, nous avons la présence de corridors d'importation et d'exportation, aboutissant ou provenant de plusieurs ports brésiliens. Il se produit un phénomène de congestion dans la partie du réseau correspondant à ces corridors. L'accent est mis sur l'étude du flot de marchandises dans ces corridors. Cela explique d'autant plus l'importance des ports dans le réseau global. Il s'agit donc de modéliser les ports pour reproduire les activités importantes qui s'y déroulent à un niveau de détail suffisant pour une planification stratégique à l'échelle nationale.

Par conséquent, dans un premier temps (Chapitre 1), nous décrivons le problème de calcul de flot maximal multiproduit. Après avoir situé le problème auquel nous nous attaquons, nous donnons les caractéristiques des différentes méthodes de résolution proposées jusqu'à maintenant. Vient ensuite un exposé mathématique des différentes constituantes de la méthode retenue qui est la méthode du lagrangien augmenté ainsi que les détails de la technique de résolution. Nous exposons des extensions au modèle et nous présentons les résultats numériques effectués sur des réseaux tests de petite taille .

Dans un deuxième temps (Chapitre 2), nous traitons du problème de modélisation des ports, soit de la représentation des ports, de la détermination de leurs caractéristiques, et de l'analyse de l'entreposage à l'intérieur d'une même période de temps ou entre deux périodes de temps consécutives.

CHAPITRE 1
CALCUL DU FLOT MAXIMAL MULTIPRODUIT

1.1 Enoncé du problème

Un des aspects de l'outil informatique pour la planification stratégique du transport de marchandises concerne la détermination du flot maximal de produits pouvant être envoyé de chaque origine vers chaque destination. Cette information est nécessaire pour pouvoir simuler correctement les flots de marchandises soit un mode de transport particulier, soit sur tous les modes à la fois, ou soit sur un sous-ensemble de modes. Nous considèrons le réseau de transport de marchandises comme un ensemble de noeuds et de liens. Le problème se prête très bien à une modélisation mathématique issue de la théorie des graphes et des réseaux. Ainsi le problème de calcul du flot maximal multiproduit dans un réseau de transport de marchandises devient, en termes mathématiques, un problème de multiflots sur un graphe orienté connexe. La résolution du problème s'effectue en trois étapes :

- modélisation mathématiques du problème.
- élaboration d'un algorithme de résolution.
- implantation informatique de la méthode.

Le problème considéré est celui du calcul du flot maximal multiproduit, étant donné que les produits pour chaque paire O/D n'ont pas la même importance, c'est-à-dire le même poids. Nous désirons maximiser la quantité envoyée de chaque origine vers chaque destination pour chaque produit, avec les contraintes de conservation de flot et une contrainte de capacité pour chaque lien (le flot total sur un lien ne doit pas dépasser la capacité de ce lien). Le problème s'écrit :

$$\text{Max} \sum_{p \in P} \sum_{i \in I} g_i^p \gamma_i^p$$

$$\text{sujet à :} \quad \sum_{k \in K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p \in P \text{ et pour tout paire O/D } i \in I$$

$$h_k \geq 0, \quad k \in K_i^p \qquad \text{pour tout produit } p \in P \text{ et pour tout paire O/D } i \in I$$

$$\sum_{p \in P} f_a^p \leq Cap_a \qquad a \in A$$

$$f_a^p \geq 0 \qquad a \in A,\ p \in P$$

où :

I	est l'ensemble des paires origine/destination (O/D) du réseau
A	est l'ensemble des liens (arcs) du réseau
P	est l'ensemble des produits
g_i^p	est la demande du produit p pour la paire O/D i
γ_i^p	est le poids du produit p pour la paire O/D i
$h_k,\ k \in K_i^p$	est le flot du produit p sur le chemin k
K_i^p	est l'ensemble des chemins reliant l'origine et la destination de la paire O/D i pour le produit p
Cap_a	est la capacité du lien a
f_a^p	est le flot du produit p sur le lien a
f_a	est le flot total sur le lien a

On a évidemment les relations suivantes :

$$f_a^p = \sum_{i \in I} \sum_{k \in K_i^p} \delta_{ak} h_k \quad \text{et} \quad f_a = \sum_{p \in P} f_a^p$$

où $\delta_{ak} = 1$ si l'arc a appartient au chemin k

$\delta_{ak} = 0$ sinon

Il est à noter que le modèle tel qu'énoncé ci-haut est présenté sous une forme conceptuelle, et que la notion de chemins introduite dans les contraintes n'est utilisée que dans un but de simplification de

la formulation. Les chemins ne seront jamais considérés de manière explicite. Ces contraintes permettent de refléter la conservation de flots aux sommets. La contrainte de non-négativité du flot f_a^p pour chaque arc $a \in A$ et pour chaque produit $p \in P$ est redondante parce que si $h_k \geq 0$, $k \in K_i^p$, $p \in P$, $i \in I$, alors cela implique que $f_a^p \geq 0$, $a \in A, p \in P$.

L'aspect multiproduit complique le problème. Les produits interagissent quand ils circulent sur les mêmes arcs soit parce qu'ils sont en compétition à cause de la capacité de l'arc, ou soit à cause des effets de congestion.

Les contraintes de flot décrivent le mouvement de la marchandise de leur point de départ à leur point d'arrivée. Les fonctions convexes décrivent les coûts dus à la congestion et aux distances parcourues. Le premier principe de Wardrop stipule que puisque chaque "usager" cherche à minimiser ses coûts de déplacements, tous les chemins utilisés par les "usagers" (soit les marchandises) de même origine et destination auront le même coût dans la réalité et tous les chemins non-utilisés auront un plus grand coût.

1.2 Revue de littérature

Dans cette section, nous présentons une revue des méthodes de résolution du problème de calcul du flot maximal multiproduit recencées dans la littérature. Le modèle de calcul du flot maximal multiproduit a de nombreuses applications dans le domaine des transports et des communications.

Le problème de calcul du flot maximal multiproduit traite la quantité de flot requise g_i^p comme une variable de décision et cherche à maximiser la somme totale pondérée compte tenu des contraintes de conservation de flot et de capacité. C'est un problème particulier du problème de multiflot à coût minimum.

Le problème de calcul du flot maximal multiproduit découle naturellement du problème de calcul

du flot maximal à un seul produit. L'efficacité des méthodes pour résoudre le problème dans le cas d'un seul produit et l'aspect théorique intéressant du théorème d'une coupe minimale (le flot maximal est égal à la capacité de la coupe minimale) ont poussé les chercheurs à essayer de trouver une généralisation pour le cas multiproduit. Malheureusement cette généralisation ne préserve pas les propriétés du cas d'un seul produit. En effet, une coupe minimale (ou un ensemble de déconnexion) est un ensemble d'arcs tel que si on enlève les arcs de cette coupe, il n'existe plus aucune chaîne de la source à la destination pour tout produit p. Une coupe minimale a une capacité minimale. Par dualité faible, nous avons toujours que le flot maximal est inférieur à la coupe minimale et un réseau est dit "sans écart" si on a l'égalité. Les problèmes à un seul produit sont "sans écart" à cause du théorème de flot maximal/coupe minimale mais l'inégalité peut être stricte dans le cas de plusieurs produits. Il a été aussi prouvé que le problème à plusieurs produits peut ne pas donner des solutions entières dans le cas où les capacités sur les arcs sont des entiers.

Dans le cas de plusieurs produits, le problème semble perdre la plupart des propriétés qui caractérisent les cas à un seul produit. Par conséquent, traiter ces problèmes comme des systèmes bloc-angulaires (avec une structure de réseau dans chaque bloc) semble être la meilleure méthode.

Certains types de problèmes particuliers ont été étudiés en profondeur : c'est le cas du problème de calcul du flot maximal à deux produits (p=2). Des algorithmes de résolution de ce type de problème ont été fournis par Hu, Rothschild et Whinston (Rothschild et al. [1966]), par Sakarovitch (Sakarovitch [1973]). Il est difficile d'étendre les résultats trouvés à des problèmes à trois produits ou plus, sauf dans des cas très spécifiques.

Les techniques de résolution du problème de calcul du flot maximal multiproduit peuvent utiliser plusieurs approches de base, dont : la décomposition, (par allocation des prix ou des ressources), le partitionnement, et la méthode du lagrangien augmenté. Ces méthodes sont employées dans le développement de méthodes spécialisées de problèmes de flot dans les réseaux à plusieurs produits.

1.2.1 Décomposition basée sur l'allocation des prix

Une décomposition basée sur l'allocation des prix est caractérisée par le changement de fonctions objectives (soit les prix) de certains sous-problèmes du problème principal. L'objectif consiste à obtenir un ensemble de prix (ou de variables duales) tel que la solution combinée de tous les sous-problèmes donne une solution optimale pour le problème original. Plusieurs études ont été proposées sur une variation de ce thème.

La décomposition usuelle de Dantzig-Wolfe peut être considérée comme une décomposition basée sur les prix étant donné que l'information du problème principal est passée aux sous-problèmes sous la forme de prix.

Une de ces études du calcul du flot maximal multiproduit a été faite par L.R. Ford, et D.R. Fulkerson (Ford et Fulkerson [1958]). La méthode proposée consiste à utiliser la méthode du simplexe appliquée à la formulation arc-chemin du problème. Pour cette formulation particulière, la matrice du programme linéaire est la matrice d'incidence des arcs versus toutes les chaînes des origines aux destinations pour tous les produits. Puisque le nombre de variables dans cette formulation est trop grand pour être traité explicitement, la méthode considère les variables hors-base implicitement en remplaçant la méthode usuelle de détermination du prochain vecteur à entrer dans la base par plusieurs combinaisons d'algorithmes combinatoires pour trouver le plus court chemin reliant une paire de noeuds dans le réseau. Les problèmes à résoudre étant très grands en taille, la méthode de génération de colonnes proposée par Ford & Fulkerson convient mieux à la résolution de ce genre de problèmes que d'autres méthodes telles la méthode du simplexe. Cette méthode est basée sur la décomposition de Dantzig-Wolfe. Nous utilisons le théorème de Goldman qui établit qu'un polytope peut s'écrire comme la somme de l'enveloppe convexe de ses points extrêmes et du cône engendré par ses rayons extrêmes.

L'application du résultat de ce théorème nous permet de construire la formulation arc-chemin du modèle. Dans la résolution de ce nouveau modèle par le simplexe révisé, la détermination de la variable d'entrée correspond à la résolution d'un problème de plus court chemin.

Formulation arc-chemin :

Soit $a_1, a_2, \ldots, a_m$ une liste des arcs dans le réseau et $c_1, c_2, \ldots, c_k$ une liste de tous les chemins reliant, pour tous les produits, toutes les origines à toutes les destinations d'un certain produit p. Soit (A_{rs}) la matrice d'incidence m x k des liens versus les chemins de produit.

$$A_{rs} = \begin{cases} 1 & \text{si } c_s \text{ contient } a_r \\ 0 & \text{sinon} \end{cases}$$

Soit x_s , s=1...k la quantité de flot de produit sur le chemin c_s et Cap_r la capacité du lien a_r ; alors le problème de flot maximal multiproduit est un problème linéaire. On suppose que toutes les paires O/D pour tous les produits p ont le même poids $\gamma_i^p = 1$.

$$\text{Max} \sum_{s=1}^{k} x_s$$

$$\text{sujet à : } \sum_{s=1}^{k} A_{rs} x_s + x_{k+s} = Cap_r \quad \text{pour tout } r$$

$$x_1, x_2, \ldots, x_{k+r} \geq 0$$

Puisque le nombre de chaînes est normalement très grand, en pratique on utilise la méthode du simplexe révisé qui consiste à garder uniquement la base $B=(b_{rj})$ ou son inverse, une matrice carrée d'ordre m et à calculer les multiplicateurs du simplexe λ_r (r=1....m) satisfaisant pour chaque $j=s_1 \ldots s_k$

$$\sum_{r=1}^{m} \lambda_r b_{rj} = \begin{cases} 1 & \text{si } j \leq k \\ 0 & \text{si } j > k \end{cases}$$

La méthode de génération de colonne pour trouver le vecteur entrant dans la base (ou pour prouver que

la base actuelle est optimale) consiste à trouver le chemin le plus court.

Un algorithme pour déterminer le plus court chemin :

Les multiplicateurs λ_r peuvent être considérés comme les coûts des arcs. Nous désirons trouver une chaîne c_s, si elle existe, de longueur $\sum_{r=1}^{m} \lambda_r A_{rs}$ plus petite que un, le coefficient de x_s dans la fonction objective. Pour ce faire, il suffit de trouver, pour chaque produit, un plus court chemin des origines vers les destinations pour ce produit. Si chaque chemin a une longueur d'au moins un, la base est optimale. Sinon, une colonne de la matrice d'incidence correspondant à une des chaînes de longueur inférieure à un peut être introduite dans la base. Le vecteur de sortie de la base est déterminé de la façon habituelle dans la méthode du simplexe révisé. Remarquons qu'il existe de nombreux algorithmes pour déterminer le plus court chemin. Notons également qu'à chaque itération, la résolution d'un problème de plus court chemin pour chaque produit p peut fournir une borne inférieure sur la valeur optimale.

1.2.2 Décomposition basée sur l'allocation des ressources

La décomposition basée sur l'allocation des ressources consiste à distribuer la capacité des arcs parmi les produits individuellement de telle manière que la résolution de |P| sous-problèmes apporte la solution optimale du problème original. A chaque itération, on alloue les ressources, soit les diverses capacités et |P| problèmes de flot à un seul produit sont résolus. La somme des capacités allouées à un arc pour tous les produits est égale à la capacité de l'arc du problème original. Par conséquent, les flots combinés des solutions de tous les sous-problèmes donnent une valeur du flot réalisable pour le problème original. On teste si la solution courante est optimale et la procédure se termine, ou une nouvelle allocation de la capacité des arcs est générée.

Modèle d'allocation :

Allouons à chaque flot une partie $SCap^p$ de la capacité totale. Le vecteur $SCap_a{}^p$ dénote la capacité allouée pour le flot du produit $p \in P$ sur l'arc a. La contrainte $\sum_{p \in P} f_a{}^p \leq Cap_a$ peut alors être décomposée en p contraintes : $f_a{}^p \leq SCap_a{}^p$. Nous obtenons le modèle suivant :

$$\text{Max} \sum_{p \in P} V^p (SCap^p)$$

$$\text{sujet à :} \sum_{p \in P} SCap^p = Cap$$

$$SCap^p \geq 0 \qquad \text{pour tout produit } p \in P$$

où le sous-problème pour un produit p donné est :

$$V^p (SCap^p) = \text{Max} \sum_{i \in I} g_i{}^p \gamma_i{}^p$$

$$\text{sujet à :} \sum_{k \in K_i{}^p} h_k = g_i{}^p \qquad \text{pour toute paire O/D } i \in I$$

$$h_k \geq 0,\ k \in K_i{}^p \qquad \text{pour toute paire O/D } i \in I$$

$$f_a{}^p \leq SCap_a{}^p \qquad a \in A$$

où $V^p (SCap^p)$ est la valeur du flot maximal étant donné une allocation de capacité $SCap^p$ sur le réseau et où Cap^p et $SCap^p$ sont des vecteurs de dimension $|A|$.

Pour un vecteur de capacité $SCap^p$ fixé, $V^p (SCap^p)$ se calcule en résolvant un problème de flot maximal à un seul produit, pour chaque produit $p \in P$. La méthode de résolution pour $SCap^p$ fixé peut être la méthode du simplexe ou la méthode des arcs non-conformes ("out-of-kilter"). Chacun de ces problèmes est un problème de flot à un seul produit avec contraintes de capacité. Pour être conséquent, notre allocation ne doit pas dépasser la capacité totale disponible, c'est-à-dire nous devons avoir que : $\sum_{p \in P} Cap_a{}^p \leq Cap_a$ pour tout $a \in A$.

Références :

Le modèle d'allocation de ressources a été utilisé par Robacker (Robacker [1956]), Sakarovitch (Sakarovitch [1969]), Swoveland (Swoveland [1971]), Held,Wolfe et Crowder (Held et al. [1974]), Kennington et Shalaby (Kennington et al. [1976]).

Sakarovitch (Sakarovitch [1969]) a donné une caractérisation duale dans le cas où les solutions optimales aux sous-problèmes à un seul produit constituent une solution optimale pour le système global. Il utilise le concept de flot de blocage. Un flot de blocage est un flot tel qu'aucune chaîne d'augmentation de flot n'existe pour aucun produit p lorsque les flots des autres produits sont fixés. Un tel flot définit un ensemble de déconnection. A chaque itération, un flot de blocage peut être obtenu en résolvant les sous-problèmes à un seul produit pour chacun des produits. Si les flots à plusieurs produits résultant ne satisfont pas le critère d'optimalité que Sakarovitch développe pour les flots de blocage, une phase d'échange de flot est exécutée au cours de laquelle des changements dans les flots sont effectués pour accroître le flot total et déterminer un nouveau flot de blocage. Pour ce faire, un système linéaire résultant de la minimisation d'une forme quadratique doit être résolue.

Swoveland (Swoveland [1971]), pour sa part, a expérimenté la méthode de linéarisation externe. Held, Wolfe et Crowder (Held et al. [1974]), ont expérimenté la méthode des sous-gradients pour résoudre le problème principal.

La méthode du sous-gradient s'applique à l'optimalité de fonctions sous-différentiables. Elle génère une suite de points de façon itérative en utilisant l'information fournie par un sous-gradient au point courant. Le problème principal peut être reformulé de la façon suivante :

$$\text{Max} \quad V(SCap) = \sum_{p \in P} V^p(SCap^p)$$

$$\text{sujet à :} \quad \sum_{p \in P} SCap^p = Cap$$

$$SCap^p \geq 0, \qquad \text{pour tout produit } p \in P$$

où V (SCap) peut être considéré comme une fonction de tous les $SCap^p$ pour $p \in P$ et

où V^p ($SCap^p$) est la valeur du flot maximal étant donné une allocation de capacité $SCap^p$ sur le réseau.

Soit Q^p l'ensemble de toutes les coupes dans le réseau G^p du produit p, c'est-à-dire les ensembles de déconnection d'arcs des chaînes de l'origine s^p à la destination t^p.

Chaque coupe $q^p \in Q^p$ peut être identifiée par un vecteur α^p dont les composantes sont:

$$\alpha_{ij}^p = 1 \quad \text{si } (i,j) \in q^p \quad \text{et} \quad \alpha_{ij}^p = 0 \quad \text{sinon.}$$

Soit R^p l'ensemble de tous les vecteurs correspondant à ces coupes.

Alors par le théorème de flot maximal / coupe minimale pour un seul produit,

$$V^p(SCap^p) = \underset{q^p \in Q^p}{\text{Min}} \left\{ \sum_{(i,j) \in q^p} Scap_{ij}^p \right\} = \underset{\alpha^p \in R^p}{\text{Min}} \; \alpha^p \, SCap^p$$

Soit $\alpha = (\alpha^1, \alpha^2, \ldots\ldots, \alpha^p)$ et $SCap = (SCap^1, SCap^2, \ldots\ldots, SCap^p)$.

En utilisant les égalités écrites ci-dessous, le problème principal peut être reformulé de la façon suivante:

$$\text{Max } V(SCap) = \underset{\alpha^p \in R^p}{\text{Min}} \sum_{p \in P} \alpha^p \, SCap^p = \underset{\alpha^p \in R^p}{\text{Min}} \; \alpha \, . \, Scap$$

$$\text{sujet à : } \sum_{p \in P} SCap^p = Cap$$

$$SCap^p \geq 0, \qquad \text{pour tout produit } p \in P$$

Ce problème est le nouveau problème principal avec |P| sous-problèmes de flot maximal à un seul produit. Etant donné une allocation de ressources : $\overline{SCap_l} = (\overline{SCap_l}^1, \overline{SCap_l}^2, \ldots, \overline{SCap_l}^p)$ à une itération l, pour chaque produit p, on résout un problème de flot maximal avec ces capacités fixées par un algorithme d'étiquettage pour obtenir le vecteur de la coupe optimale $\alpha^p(SCap_l^p)$. Ces vecteurs

sont utilisés pour construire une nouvelle allocation des capacités :

$$SCap_{l+1} = P_{SC}\left(\overline{SCap_l} + t_l \cdot \alpha\,(SCap_l) \right)$$

P_{SC} dénote la projection sur l'ensemble convexe SC pour tout SCap satisfaisant

$$\sum_{p \in P} SCap^p = Cap \text{ et } SCap^p \geq 0 \text{ , pour tout produit p.}$$

Cette projection est facile à calculer grâce à la structure simple de l'ensemble SC. Avec une séquence de pas scalaires $\{ t_l \}$ choisis convenablement, la méthode convergence vers une bonne solution SCap du problème principal.

Kennington et Shalaby (Kennington et al. [1976]) et Assad (Assad [1978]) ont utilisé cette approche pour résoudre respectivement le problème de transport à capacité limitée pour plusieurs produits et des problèmes à coût minimal. Kennington et Shalaby utilisent une technique heuristique qui donne de bonnes solutions à de larges problèmes de flots dans des réseaux à plusieurs produits. L'approche générale consiste à allouer des capacités sur les arcs pour chaque produit individuellement et à décomposer le problème en des ensembles de sous-problèmes à un seul produit. Tous les sous-problèmes à un seul produit sont résolus. La combinaison des solutions obtenues est comparée à une borne inférieure. Si la solution est dans une marge acceptable de la borne inférieure, la procédure se termine, sinon les capacités sur les arcs sont réaffectées et les sous-problèmes sont résolus à nouveau. La réallocation des capacités sur les arcs est basée sur une approche d'optimisation du sous-gradient. Une méthode simple est utilisée pour projeter cette affectation sur la région réalisable. Les besoins en mémoire sont modestes et presque toutes les opérations sont effectuées directement sur des réseaux à un seul produit.

Assad présente des résultats numériques aux problèmes linéaires de flots multiproduits avec

deux algorithmes. Le premier est une décomposition basée sur l'allocation des prix : c'est une décomposition Dantzig-Wolfe pour les systèmes blocs-angulaires. Le deuxième est une décomposition basée sur l'allocation des ressources : on optimise le sous-gradient. Assad a noté que la performance du premier algorithme est notablement supérieure.

Grinold (Grinold [1968]) a utilisé une approche théorique de jeu avec deux joueurs appelés allocation et prix. Le jeu implique les contraintes reliant les différents problèmes de flots ensemble, soit $\sum_{p \epsilon P} f_a^p \leq Cap$. L'idée fondamentale est d'allouer les capacités aux différents produits et de résoudre indépendamment les problèmes de flot pour chaque produit. A chaque itération, le premier joueur détermine une allocation $SCap^1,...SCap^p$ des capacités. La contre-stratégie du joueur 'prix' consiste à résoudre |P| problèmes de flot maximal avec les variables duales obtenues. Les variables duales servent à la stratégie des prix. Le joueur 'allocation' réplique en changeant la répartition des capacités tout en ayant que : $\sum_{p \epsilon P} SCap^p = Cap$ et $SCap^p \geq 0 \quad \forall\ p \epsilon P$.

La convergence de cet algorithme n'est pas rapide.

Gratzer et Steiglitz (Gratzer et al. [1972]) présentent une approche heuristique pour de grands problèmes de flot à plusieurs produits. La méthode cherche à obtenir une allocation $(SCap^1,....,SCap^p)$ presque optimale. A chaque itération, l'allocation précédente est changée d'abord en distribuant une partie de la capacité résiduelle parmi les produits qui saturent leur capacité puis en résolvant les problèmes de flot maximal à un seul produit. Si le flot n'a pas changé de façon significative, le flot est décrémenté dans certains chemins, et cela réduit l'augmentation de flot possible dans d'autres chemins. Pour éviter de faire trop d'effort pour localiser ces chemins, on ne cherche que ceux ayant un certain nombre de noeuds. Les valeurs du flot à 85% ou à 95% de la solution optimale peuvent être obtenus avec cet heuristique.

1.2.3 Approche de Partitionnement

Ces approches sont des cas particuliers de la méthode du simplexe pour lesquelles la base courante est partitionnée dans le but d'exploiter sa structure particulière. Ces méthodes sont des techniques spéciales des méthodes du simplexe primal, dual, ou primal-dual.

La partition primale de Rosen est une spécialisation de l'approche de relaxation. Une base est choisie pour chaque flot et les contraintes de positivité des variables de base sont relaxées. Le programme linéaire résultant a des propriétés intéressantes. Chaque colonne est le vecteur caractéristique du cycle fondamental formé par un arc hors-base avec la base. Le modèle compte seulement une contrainte par arc, ce qui explique l'intérêt de la méthode.

Grigoriadis et White (Grigoriadis et al. [1972]) ont présenté un algorithme efficace pour le problème de flot multiproduit basé sur un algorithme de partitionnement de Rosen pour les systèmes présentant une structure bloc angulaire. Des expériences de calcul ont montré que seulement quelques contraintes de capacité sont contraignantes en pratique et donc la plupart de ces contraintes peuvent être relaxées sans changer l'optimum. Cette observation motive l'idée de diviser les contraintes de capacité en deux groupes : les contraintes courantes (contrainte saturées ou presque saturées) et secondaires. Le nouveau problème est le suivant :

$$\text{Max} \quad \sum_{i \in I} \sum_{p \in P} g_i^p \gamma_i^p$$

$$\text{sujet à :} \quad \sum_{k \in K_i^p} h_k = g_i^p \qquad \text{pour toute paire O/D } i \in I \text{ et pour tout produit } p \in P$$

$$h_k \geq 0, \; k \in K_i^p \qquad \text{pour toute paire O/D } i \in I \text{ et pour tout produit } p \in P$$

$$\sum_{p \in P} f_a^p \leq Cap_a \qquad \text{pour } a \in A'$$

où A' est un sous-ensemble de A des contraintes courantes.

La méthode de partitionnement peut alors être appliquée à ce nouveau problème.

1.2.4 Méthode du lagrangien augmenté

Cette approche est par essence duale, car elle fournit une suite de valeurs non-réalisables. Elle comprend deux composantes essentielles :

(a) la règle de l'ajustement des multiplicateurs associés aux contraintes de capacité relaxée.

(b) la résolution d'un sous-problème de multiflot d'équilibre à coûts convexes.

Dans sa thèse sur les problèmes de multiflot compatible à coût minimum, M. Sauvé (Sauvé [1984]) indique que la méthode du Lagrangien Augmenté est plus intéressante que d'autres méthodes: par exemple, au niveau des besoins en mémoire centrale, la méthode du Lagrangien Augmenté est meilleure que certaines méthodes telles la méthode de Dantzig-Wolfe ou la méthode de partitionnement de Rosen. De plus, étant donné que la méthode peut fonctionner à partir de la valeur courante d'un flot et qu'elle nécessite uniquement la connaissance du flot total sur chaque arc, elle permet la résolution de problèmes de grande taille. Cette méthode est décrite plus en détail à la prochaine section.

1.3 Approche lagrangienne augmentée

La méthode utilisée pour résoudre ce problème est la méthode du **Lagrangien Augmenté** ou méthode des multiplicateurs. Cette méthode combine méthode de pénalité et relaxation lagrangienne. Les deux aspects combinés permettent d'éliminer plusieurs désavantages associés à l'une ou à l'autre prise séparément. On peut considérer la méthode des multiplicateurs comme l'application de la relaxation lagrangienne à un modèle dont la fonction objective a été augmentée d'un terme de pénalité. Cette approche est par essence duale car elle fournit une suite de valeurs non-réalisables. Les contraintes de capacité sont relaxées et la fonction objective est augmentée d'un terme de pénalité.

1.3.1 Forme générale du lagrangien augmenté

L'analyse qui suit est extraite du chapitre 13 du livre de David G. Luenberger (Luenberger) sur la programmation linéaire et non-linéaire. Elle a été reportée ici dans le but d'expliquer la forme du lagrangien augmenté dans le cas du problème de flot maximal multiproduit.

1.3.1.1 Contraintes d'égalité

Le lagrangien augmenté pour les problèmes avec contraintes d'égalité de la forme :

$$\text{Min} \quad f(x)$$
$$\text{sujet à : } h(x)=0 \qquad (1)$$

est la fonction $L_{pen}(x,\lambda) = f(x) + \lambda^t . h(x) + 1/2 . pen . | h(x) |^2$ pour une constante **pen** positive.

Le lagrangien peut être considéré sous deux points de vue, celui de la fonction de pénalité et celui de la dualité.
Du point de vue de la fonction de pénalité, le lagrangien augmenté, pour une valeur fixée du vecteur de multiplicateur λ, est simplement la fonction quadratique standard de pénalité du problème :

$$\text{Min } f(x) + \lambda^t . h(x)$$
$$\text{sujet à : } h(x) = 0 \qquad (2)$$

Ce problème est équivalent au problème original **(1)** car n'importe quelle combinaison des contraintes ajoutées à la valeur de l'objectif original n'affecte en rien la solution optimale ni la valeur de la fonction objective à l'optimum. Remarquons que si le vecteur de multiplicateurs est égal à λ^*, le bon vecteur de multiplicateurs, alors le lagrangien augmenté semble être une fonction de pénalité exacte.

A chaque étape k de la méthode du lagrangien augmenté, on considère un vecteur de multiplicateurs fixé λ_k. La solution x_k est le minimum de :

$$f(x) + \lambda_k^t . h(x) + 1/2 . pen . | h(x) |^2 \qquad (3)$$

Le prochain λ_k est déterminé de la façon standard suivante : $\lambda_{k+1} = \lambda_k + pen . h(x_k)$

Pour justifier cette procédure d'ajustement, considérons le problème **(2)** avec $\lambda = \lambda_k$. Les multiplicateurs lagrangiens associés à ce problème sont : $\lambda^* - \lambda_k$, où λ^* est le vecteur de multiplicateur de **(1)**. D'un autre côté, **(3)** étant la fonction de pénalité correspondant à **(2)**, il s'en suit que $pen . h(x_k)$ est approximativement égal aux multiplicateurs de **(2)**. Nous obtenons que : $pen . h(x_k) \equiv \lambda^* - \lambda_k$. Donc une bonne approximation du vecteur de multiplicateurs inconnu λ^* est : $\lambda_{k+1} = \lambda_k + pen . h(x_k)$.

Si le paramètre de pénalité **pen** est suffisamment élevé, le lagrangien augmenté a un point minimum local près du point optimal réel. Par continuité, le résultat peut s'étendre au voisinnage de x^*, λ^* : c'est-à-dire pour tout λ près de λ^*, le lagrangien augmenté a un point minimum local unique près de x^*. Cette correspondance entre x et λ définit une fonction continue. Si on peut déterminer λ tel que $h(x(\lambda)) = 0$, alors λ peut en fait être λ^* puisque $x(\lambda)$ satisfait les conditions nécessaires du problème original. Donc déterminer le bon vecteur de multiplicateurs λ revient à résoudre l'équation $h(x(\lambda))=0$. Pour cette raison, le processus itératif $\lambda_{k+1} = \lambda_k + pen . h(x(\lambda))$ est une méthode d'approximation successive, qui converge linéairement au voisinnage de λ^*.

Bien qu'à chaque itération de la méthode du lagrangien augmenté ce qui doit être principalement ajusté est le vecteur des multiplicateurs λ, le paramètre de pénalité **pen** peut être également ajusté. Comme dans toute méthode avec fonction de pénalité, la séquence

de paramètres **pen** est choisie à l'avance : **pen** est soit maintenu fixé, soit augmenté jusqu'à une certaine valeur finie, ou bien tend (lentement) vers l'infini. Puisque dans cette méthode, il n'est pas nécessaire pour **pen** de tendre vers l'infini (**pen** peut avoir à chaque itération une valeur relativement petite), le mauvais conditionnement dû à la fonction de pénalité est moyen.

Du point de vue de la théorie de la dualité, le lagrangien augmenté est simplement le lagrangien augmenté standard du problème :

$$\text{Min } f(x) + 1/2 \,.\, pen \,.\, |\, h(x)\,|^2$$

$$\text{sujet à : } h(x) = 0 \qquad \textbf{(4)}$$

Ce problème est équivalent au problème original **(1)** car l'addition du terme $1/2 \,.\, pen \,.\, |h(x)|^2$ à la fonction objective n'affecte en rien ni la solution optimale, ni la valeur optimale de la fonction objective, ni les multiplicateurs de Lagrange.

L'itération de base de la méthode du lagrangien augmenté concerne le vecteur de multiplicateurs λ, et donc il est naturel de considérer la méthode du point de vue de la dualité.

Le lagrangien du problème original peut ne pas être convexe près de la solution et par conséquent la théorie de la dualité ne peut être appliquée; par contre le terme $1/2 \,.\, pen \,.\, |h(x)|^2$ tend à rendre le langrangien convexe. Pour un paramètre **pen** suffisamment élevé, le lagrangien sera effectivement localement convexe. Donc la théorie de la dualité peut être employée, et le problème dual correspondant peut être résolu par un procédé itératif en λ.

Définissons la fonction duale associée au problème **(4)** :

$$\Phi(\lambda) = \min \{\, f(x) + \lambda^t \,.\, h(x) + 1/2 \,.\, pen \,.\, |\, h(x)\,|^2 \,\} \qquad \textbf{(5)}$$

dans une région du voisinnage de x^*, λ^*.

Si $x(\lambda)$ est le vecteur minimisant le terme de droite de **(5)**, $h(x(\lambda))$ est le gradient de Φ.

Donc le processus itératif $\lambda_{k+1} = \lambda_k + pen \,.\, h(x(\lambda_k))$ de la méthode du lagrangien augmenté peut être considéré comme une itération dans la direction de montée pour maximiser la fonction

duale Φ. C'est une forme simple de direction de montée, utilisant un pas constant **pen** . Si le terme de pénalité est incorporé à un problème , le problème primal est mal conditionné lorsque pen $\to \infty$ par contre le problème dual devient bien conditionné.

1.3.1.2 Contraintes d'inégalité

On peut facilement traiter les contraintes d'inégalité dans les méthodes du lagrangien augmenté. Soit le problème avec contraintes d'inégalité suivant :

$$\text{Min } f(x)$$
$$\text{sujet à : } g(x) \leq 0 \qquad \text{où } g \text{ est de dimension m.}$$

On suppose que ce problème a une solution bien définie x^*. En ajoutant des variables d'écart à chacune des contraintes, le problème peut être reformulé de la façon suivante :

$$\text{Min } f(x)$$
$$\text{sujet à : } g_j(x) + v_j = 0, \quad j=1,2,...m.$$
$$v_j \geq 0, \quad j=1,2,...m.$$

On peut appliquer la théorie des problèmes avec contraintes d'égalité.

La fonction duale correspondant à la méthode du lagrangien augmenté est :

$$L_{pen}(x, \lambda) = \min_{v \geq 0, x} \{ f(x) + \sum_{j=1}^{m} \{ \lambda_j [g_j(x) + v_j] + 1/2 \cdot pen \cdot | g_j(x) + v_j |^2 \} \}$$
$$= \min_{v \geq 0, x} \{ f(x) + \lambda^t [g(x) + v] + 1/2 \cdot pen \cdot | g(x) + v |^2 \}$$

La minimisation par rapport à v peut être déduite analytiquement, et on obtient une fonction duale impliquant une minimisation uniquement sur x.

Soit $P_j = \lambda_j [g_j(x) + v_j] + 1/2 \cdot pen \cdot [g_j(x) + v_j]^2$ pour $j=1,2,...m$

P_j est la partie de la fonction $L_{pen}(x, \lambda)$ contenant v_j.

On doit minimiser P_j par rapport à $v_j \geq 0$. Par différentiation, on obtient la solution :

$$v_j = \begin{cases} - g_j(x) - \lambda_j / pen & \text{si } (- g_j(x) - \lambda_j / pen) \geq 0 \\ 0 & \text{sinon} \end{cases}$$

qui est équivalent à : $v_j = \max [0 ; - g_j(x) - \lambda_j / pen]$ pour j=1,2,...m.

Remplaçons v_j dans l'expression P_j :

$$P_j = \begin{cases} 1 / (2.pen) . \{ [\lambda_j + pen . g_j(x)]^2 - \lambda_j^2 \} & \text{si } v_j = 0 \\ - \lambda j^2 / (2.pen) & \text{si } v_j = - g_j(x) - \lambda_j / pen \end{cases}$$

qui est équivalent à : $P_j = 1/(2.pen) . \{ [\max (0; \lambda_j + pen . g_j(x))]^2 - \lambda_j^2 \}$

Pour un vecteur de multiplicateurs $\lambda > 0$ fixé, la fonction duale devient :

$$L_{pen}(x, \lambda) = \min_x \{ f(x) + 1 / (2.pen) . \sum_{j=1}^{m} \{ [\max(0; \lambda_j + pen . g_j(x))]^2 - \lambda_j^2 \}$$

1.3.2 <u>Lagrangien augmenté appliqué au problème du flot maximal multiproduit</u>

Le problème de flot maximal multiproduit peut être reformulé de la façon suivante :

$$\text{Min} - \sum_{p \in P} \sum_{i \in I} g_i^p \gamma_i^p$$

$$\text{sujet à : } \sum_{k \in K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$h_k \geq 0, \ k \in K_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$\sum_{p \in P} f_a^p \leq Cap_a \qquad a \in A$$

En se basant sur les résultats énoncés dans le paragraphe précédent,la fonction du lagrangien augmenté est : $L_{pen}(f_a,\lambda) = - \sum_{p \in P} \sum_{i \in I} g_i^p \gamma_i^p + 1/(2.pen). \sum_{a \in A} [\max\{0;\lambda_a + pen.(\sum_{p \in P} f_a^p - Cap_a)\}^2 - \lambda_a^2]$

avec **pen** > 0 **pen** est le facteur de pénalité

$\lambda = (\lambda_a)_{a \in A}$ est le vecteur des multiplicateurs de dimension | A |

On peut écrire le modèle de la façon suivante :

PL (λ) $\quad L_{pen}(\lambda) = \text{Min } L_{pen}(f_a, \lambda)$

sujet à : $\sum_{k \in K_i^p} h_k = g_i^p$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$

$h_k \geq 0,\ k \in K_i^p$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$

et nous avons que

PL $\quad L^* = \underset{\lambda \geq 0}{\text{Max}}\ L_{pen}(\lambda)$

L'algorithme proposé par **Bertsekas** (Bertsekas [1982]) pour résoudre le problème **PL** est le suivant :

étape 0 - (Initialisation)

$\ell = 0$ (ℓ est le compteur d'itérations)

$\lambda_\ell = 0$

étape 1 - (Résolution du sous-problème à coûts convexes)

Calculer $(f_{a,\ell}^p\ ;\ g_{i,\ell}^p)$ la solution optimale de **PL** (λ_ℓ)

étape 2 - (Ajustement des multiplicateurs)

$$\lambda_{a,\ell+1} = \max\{0\ ;\ \lambda_{a,\ell} + pen_\ell \cdot (\sum_{p \in P} f_{a,\ell}^p - Cap_a)\},\ a \in A$$

$pen_{\ell+1} \geq pen_\ell$

$\ell = \ell + 1$

allera étape 1

Analysons les propriétés de convergence de l'algorithme. La fonction du lagrangien augmenté n'est pas strictement convexe en x mais la suite de points générés $(x_\ell\ ,\ \lambda_\ell)$ converge vers (x^*, λ^*) où x^* est la solution optimale du problème original et λ^* est la solution optimale du problème **PL** (Bertsekas [1982]). La rapidité de convergence de

l'algorithme dépend de la rapidité de convergence de l'algorithme choisi pour la résolution de l'étape 1.

La méthode du lagrangien augmenté consiste à la base à ajuster les multiplicateurs à chaque étape, après la résolution du sous-problème. Or, le paramètre de pénalité **pen** peut également être modifié à chaque étape, comme il a été indiqué au paragraphe précédent. On peut soit le garder à une valeur fixe, soit l'augmenter jusqu'à une certaine valeur finie, soit le faire tendre vers l'infini. Un paramètre de pénalité **pen** très grand engendre un problème dual bien conditionné mais un problème primal mal conditionné. La convergence duale sera par conséquent rapide tandis que la convergence du sous-problème de l'étape 1 sera lente. On obtient le phénomène inverse lorsque le paramètre de pénalité **pen** est petit. Le choix du paramètre **pen** influera sur le nombre global d'itérations de la méthode versus le nombre d'itérations nécessaires pour résoudre le sous-problème. Le paramètre de pénalité peut être augmenté après quelques itérations de l'algorithme. Cela aide à se rapprocher de l'ensemble des solutions réalisables.

Critères d'arrêt :

On a plusieurs critères d'arrêt pour terminer l'algorithme.

Le premier critère concerne les multiplicateurs. On arrête l'algorithme lorsque la norme de la différence entre deux vecteurs multiplicateurs de deux itérations successives est inférieure à une certaine valeur ε_1, soit que :

$$\| \lambda_{\ell+1} - \lambda_\ell \| = \| \max \{ - \lambda_\ell ; \text{pen}_\ell . (\sum_{p \in P} f_\ell^p - \text{Cap}) \} \| < \varepsilon_1$$

où λ_ℓ est le vecteur de multiplicateurs à l'étape ℓ

f_ℓ^p est le vecteur de flot des $f_{a,\ell}^p$ (de dimension $|A|$)

Cap est le vecteur des capacités (de dimension $|A|$)

ε_1 est un paramètre de précision

De plus on a que : $\| \max \{ 0 ; \sum_{p \in P} f_{\ell}^{p} - Cap \} \| < \varepsilon_1 / pen$ lorsque le critère d'arrêt est satisfait. Le vecteur de flot f_{ℓ}^{p} est alors très près d'être réalisable puisque $\sum_{p \in P} f_{\ell}^{p}$ excède la capacité par une très petite quantité ε_1/pen.

Ce critère d'arrêt n'est pas toujours satisfait après un certain nombre d'itérations même si la solution obtenue est acceptable : ceci est dû à une grande variation des multiplicateurs d'une itération à l'autre.

On utilise deux autres critères d'arrêt, l'un sur la valeur du flot prédit f_a^p, l'autre sur la valeur de la demande prédite g_i^p.

Le critère d'arrêt sur le flot prédit à l'itération ℓ est le suivant :

$$\max_{a \in A} \left(\sum_{p \in P} f_{a,\ell}^{p} - \sum_{p \in P} f_{a,\ell-1}^{p} \right) < \varepsilon_2 * Cap_{a'}$$

où a' est l'arc tel que $\left(\sum_{p \in P} f_{a,\ell}^{p} - \sum_{p e P} f_{a,\ell-1}^{p} \right)$ est maximisé.

ε_2 est un paramètre de précision.

On arrête l'algorithme si les flots de deux itérations successives ne varient pas beaucoup.

L'autre critère d'arrêt sur la demande prédite à l'itération ℓ est le suivant :

$$\max_{i \in I, p \in P} \left(g_{i,\ell}^{p} - g_{i,\ell-1}^{p} \right) < \varepsilon_3 * g_{i',\ell-1}^{p'}$$

où i' est la paire O/D et p' est le produit tel que $(g_{i,\ell}^{p} - g_{i,\ell-1}^{p})$ est maximisé.

ε_3 est un paramètre de précision.

Le modèle du **LAGRANGIEN AUGMENTE** est une méthode comportant deux composantes essentielles : 1) la règle de l'ajustement des multiplicateurs.

2) la résolution de problèmes multiflots à coûts convexes.

Cette méthode a également la caractéristique qu'elle peut fonctionner à partir de la valeur courante d'un flot vérifiant les équations de conservation de flot puisque les contraintes de capacité sont relaxées et que la méthode ajuste les multiplicateurs pour se rapprocher de l'ensemble des solutions réalisables. La méthode des multiplicateurs nécessite uniquement la connaissance du flot total pour tous les produits sur chaque arc, que ce soit dans la fonction du lagrangien augmenté aussi bien que dans les critères d'arrêt. Donc au niveau des besoins en mémoire centrale, la méthode du lagrangien augmenté est plus intéressante et meilleure que d'autres méthodes car elle permet la résolution de problèmes de grande taille.

1.4. Technique de résolution

L'algorithme proposé par Bertsekas est utilisé pour résoudre ce problème **PL**. Pour aider à l'efficacité et à la rapidité de l'algorithme, il est important de choisir à l'étape 1 une bonne méthode de résolution du sous-problème à coûts convexes.

Le sous-problème **PL(λ)** est le suivant :

$$\text{Min} \quad - \sum_{p \in P} \sum_{i \in I} g_i^p \gamma_i^p + 1/(2.\text{pen}). \sum_{a \in A} [\max \{0 ; \lambda_a + \text{pen}_\ell . (\sum_{p \in P} f_a^p - \text{Cap}_a)\}^2 - \lambda_a^2]$$

$$\text{sujet à :} \quad \sum_{k \in K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$h_k \geq 0, \; k \in K_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

Ce sous-problème est un problème de multiflot d'équilibre. PL(λ) peut être vu comme un problème d'équilibre avec demande variable. Le sous-problème est de la forme :

$$\text{Min} \sum_{a \in A} s_a(f) - \sum_{p \in P} \sum_{i \in I} \int_0^{g_i^p} \gamma_i^p \cdot \partial g_i^p$$

$$\text{sujet à :} \quad \sum_{k \in K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$h_k \geq 0, \; k \in K_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

où $s_a(f_a)$ est la partie de la fonction relaxée.

$$s_a(f_a) = 1/(2.\text{pen}) . [\max\{0; \lambda_a + \text{pen}_\ell . (\sum_{p \in P} f_a^p - \text{Cap}_a)\}^2 - \lambda_a^2]$$

f_a est le vecteur des flots.

1.4.1 Conditions Kuhn- Tucker

Nous allons déterminer les conditions Kuhn-Tucker pour une analyse appronfondie de **PL(λ)**.

Soit Δ_i^p, $i \in I$, $p \in P$, la variable duale associée à la contrainte de conservation de flot et soit v_k,

$k \varepsilon K_i^p$, $i \varepsilon I$, $p \varepsilon P$, la variable duale associée à la contrainte de non-négativité du flot de chemin.

Les conditions Kuhn-Tucker sont :

(a) (en prenant la dérivée par rapport à la variable h_k)

$$\sum_{a \varepsilon A} \delta_{ak} \cdot \partial s_a (f_a) / \partial h_k - \Delta_i^p - v_k = 0, \quad k \varepsilon K_i^p, i \varepsilon I, p \varepsilon P$$

$$h_k \cdot v_k = 0, \quad k \varepsilon K_i^p, i \varepsilon I, p \varepsilon P$$

$$v_k \geq 0, \quad k \varepsilon K_i^p, i \varepsilon I, p \varepsilon P$$

Si le flot sur le chemin k est non nul, on a que $h_k > 0$ alors $v_k = 0$.

Et cela implique que : $\Delta_i^p = \sum_{a \varepsilon A} \delta_{ak} \cdot \partial s_a (f_a) / \partial h_k$ **(1)**

(b) (en prenant la dérivée par rapport à la variable g_i^p)

$- \gamma_i^p + \Delta_i^p = 0$ implique que $\gamma_i^p = \Delta_i^p$ **(2)**

(1) et **(2)** impliquent que $\sum_{a \varepsilon A} \delta_{ak} \cdot \partial s_a (f_a) / \partial h_k = \gamma_i^p$

A l'équilibre, on obtient donc que le poids du produit p pour la paire O/D i doit être égal à la somme sur les arcs utilisés de la dérivée de la partie de la fonction relaxée.

1.4.2 Algorithme de type PARTAN

Pour résoudre ce type de problème, on utilise une variante de l'algorithme Frank & Wolfe : la méthode PARTAN. J. Guélat (Guélat [1983]) a montré dans sa thèse que la méthode PARTAN converge plus rapidement dans la majorité des cas que l'algorithme Frank & Wolfe. L'algorithme correspondant à la méthode PARTAN est explicité en annexe A.

Dans l'étape 1 de l'algorithme utilisant la méthode PARTAN, il est nécessaire de déterminer la direction de descente Frank & Wolfe, soit de calculer la solution optimale du sous-problème linéaire du problème **PL(λ_ℓ)** pour un λ_ℓ fixé.

Notons que

la variable H_k, $k \varepsilon K_i^p$ correspond à la variable h_k, $k \varepsilon K_i^p$

la variable w_i^p correspond à la variable g_i^p

la variable F_a^p correspond à la variable f_a^p.

Le sous-problème linéaire associé au problème $PL(\lambda_\ell)$ à résoudre est le suivant :

$$\text{Min} \sum_{p \varepsilon P} \sum_{i \varepsilon I} \sum_{k \varepsilon K_i^p} \left(\sum_{a \varepsilon A} \delta_{ak} \cdot \partial s_a(f_a) / \partial h_k \right) H_k - \sum_{p \varepsilon P} \sum_{i \varepsilon I} \gamma_i^p w_i^p$$

$$\text{sujet à :} \quad \sum_{k \varepsilon K_i^p} H_k = w_i^p \qquad \text{pour tout produit } p \varepsilon P \text{ et pour toute paire O/D } i \varepsilon I$$

$$H_k \geq 0, \; k \varepsilon K_i^p \qquad \text{pour tout produit } p \varepsilon P \text{ et pour toute paire O/D } i \varepsilon I$$

La fonction objective devient à l'itération r :

$$\sum_{p \varepsilon P} \sum_{i \varepsilon I} \left(\sum_{k \varepsilon K_i^p} s'_{kr} H_k - \gamma_i^p w_i^p \right)$$

où $s'_{kr} = \sum_{a \varepsilon A} \delta_{ak} \cdot \partial s_a(f_a) / \partial h_k$ à l'itération r

s'_{kr} est le "coût" sur le chemin k à l'itération r.

Notons que pour toutes solutions w_i^p, le coût sera minimisé si w_i^p est affecté sur le chemin le plus court. Soit u_i le coût du chemin le plus court pour la paire O/D i.

La fonction objective du sous-problème de $PL(\lambda_\ell)$ devient :

$$\sum_{p \varepsilon P} \sum_{i \varepsilon I} \left(\sum_{k \varepsilon K_i^p} u_i H_k - \gamma_i^p w_i^p \right)$$

$$= \sum_{p \varepsilon P} \sum_{i \varepsilon I} \left(u_i \sum_{k \varepsilon K_i^p} H_k - \gamma_i^p w_i^p \right)$$

$$= \sum_{p \varepsilon P} \sum_{i \varepsilon I} \left(u_i w_i^p - \gamma_i^p w_i^p \right) \qquad \text{à cause de la contrainte } \sum_{k \varepsilon K_i^p} H_k = w_i^p$$

$$= \sum_{p \varepsilon P} \sum_{i \varepsilon I} \left(u_i - \gamma_i^p \right) w_i^p$$

Le problème de l'étape 1 de l'algorithme PARTAN revient donc à :

$$\text{Min} \sum_{p \in P} \sum_{i \in I} (u_i - \gamma_i^p) w_i^p$$

sujet à : $w_i^p \geq 0, \quad i \in I, p \in P$

Le sous-problème de PL (λ_ℓ) se résoud facilement puisque :

si $u_i < \gamma_i^p$, nous avons intérêt à maximiser w_i^p

si $u_i > \gamma_i^p$, nous avons intérêt à minimiser w_i^p.

La solution de ce sous-problème est obtenue en calculant les chemins les plus courts u_i pour chaque paire O/D i. Puis les demandes w_i^p sont obtenues de cette façon : pour chaque paire O/D $i \in I$ et pour chaque produit $p \in P$,

$$w_i^p = \begin{cases} g_i^{p\,max} & \text{si } u_i < \gamma_i^p \\ g_{i,r}^p & \text{si } | u_i - \gamma_i^p | < \varepsilon \\ 0 & \text{si } u_i > \gamma_i^p \end{cases}$$

La somme des flots sur un arc pour tous les produits $\sum_{a \in A} f_a^p$ est obtenue par conséquent en en affectant $\sum_{p \in P} w_i^p$ sur le plus court chemin pour toute paire O/D i.

La direction de descente est : $(d_r ; e_r) = (F_{a,r}^p ; w_{i,r}^p) - (f_{a,r}^p ; g_{i,r}^p)$

où $F_{a,r}^p = \sum_{i \in I} \sum_{k \in K_i^p} \delta_{ak} H_k$ et $\delta_{ak} = \begin{cases} 1 & \text{si l'arc a appartient au chemin k} \\ 0 & \text{sinon} \end{cases}$

Donc l'algorithme PARTAN appliqué au problème **PL(λ_ℓ)** est :

étape 0 - (Initialisation)

$(f_{a,1}^p ; g_{i,1}^p)$ une solution réalisable ; r=1

étape 1 - (Direction de descente Frank & Wolfe)

Calculer $(F_{a,r}^p ; w_{i,r}^p)$ la solution optimale de :

$\text{Min } \nabla L_{pen} (f_{a,r}^p ; \lambda) \; (F_{a,r}^p ; w_{i,r}^p)^t$

sujet à : $\sum_{k \in K_i^p} H_k = w_i^p$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$

$H_k \geq 0, k \in K_i^p$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$

$F_a^p \geq 0$ $a \in A$

Pour ce faire

1) on calcule les chemins les plus courts u_i pour tout $i \in I$

2) w_i^p est obtenu par :

$$w_i^p = \begin{cases} g_i^p \max & \text{si } u_i < \gamma_i^p \\ g_{i,r}^p & \text{si } |u_i - \gamma_i^p| < \varepsilon \\ 0 & \text{si } u_i > \gamma_i^p \end{cases}$$

3) $\sum_{p \in P} F_a^p$ pour tout $a \in A$ est obtenu en affectant $\sum_{p \in P} w_i^p$ pour tout $i \in I$ sur le plus court chemin dont le coût est u_i. La direction de descente Frank & Wolfe est donc : $d_r = (F_{a,r}^p ; w_{i,r}^p) - (f_{a,r}^p ; g_{i,r}^p)$

étape 2 - (Recherche unidimensionnelle dans la direction Frank & Wolfe)

Résoudre $\min_{0 \leq \delta \leq 1} L_{pen} ((f_{a,r}^p ; g_{i,r}^p) + \delta d_r)$

La solution optimale δ_r est le "pas Frank & Wolfe".

$z_r = (f_{a,r}^p ; g_{i,r}^p) + \delta_r d_r$

étape 3 - (Recherche dans la direction PARTAN)

si r=1 alors $z_r = (f_{a,r+1}^p ; g_{i,r+1}^p)$

sinon la direction de recherche PARTAN est :

$d_r^p = z_r - (f_{a,r-1}^p ; g_{i,r-1}^p)$

On calcule le pas maximal p^{max} dans cette direction.

On effectue une recherche unidimensionnelle dans la direction PARTAN soit, on résoud : $\min_{0 \leq p \leq p^{max}} L_{pen} ((f_{a,r-1}^p ; g_{i,r-1}^p) + p \, d_r^p)$

La solution optimale p_r est le "pas PARTAN".

$(f_{a,r+1}^p ; g_{i,r+1}^p) = (f_{a,r-1}^p ; g_{i,r-1}^p) + p_r d_r^p$

r=r+1
retour à l'étape 1

1.4.3 Détails d'implantantion

1) Il n'est pas nécessaire d'avoir en mémoire chaque flot f_a^p pour chaque produit p .Il suffit de connaître la somme des flots pour tous les produits sur chaque arc aεA.

2) L'implantation de l'algorithme nécessite la mémorisation de deux solutions les plus récentes, soit : $\sum_{p\varepsilon P} f_{a,r}^p \quad g_{i,r}^p$ et $\sum_{p\varepsilon P} f_{a,r-1}^p \quad g_{i,r-1}^p$ avec iεI et aεA

Ceci découle de l'utilisation d'un algorithme de type PARTAN pour le sous-problème d'équilibre avec demande variable.

3) Pour déterminer δ="pas Frank & Wolfe" ou p="pas Partan", on utilise un combinaison de la méthode de bissection et de la méthode régula falsi (Guélat [1983]).

4) Réoptimisation :

Pour améliorer la convergence de l'algorithme de résolution du problème PL, on peut procéder comme suit : après l'ajustement des multiplicateurs de Lagrange, il n'est pas nécessaire de recommencer avec une nouvelle solution réalisable initiale $(f_{a,1}^p ; g_{i,1}^p)$ pour résoudre le sous-problème PL(λ_ℓ) avec λ_ℓ fixé (étape 0 de l'algorithme de résolution du sous-problème). Il suffit de partir avec la solution obtenue à l'itération précédente (ℓ-1) lors de la résolution du problème PL ($\lambda_{\ell-1}$). Donc on peut modifier l'algorithme de résolution du sous-problème PL (λ_ℓ) basé sur une variante de Frank & Wolfe comme suit :

étape 0 - (Initialisation)

$(f_{a,1}^p ; g_{i,1}^p) = (f_{a,\ell-1}^p ; g_{i,\ell-1}^p)$

r = 1

1.5 Extensions du modèle

1.5.1 Flot fixé sur chaque lien

1.5.1.1 Enoncé du problème

Le problème considéré ici est plus général que celui décrit en 1.1. Il s'agit de calculer le flot maximal multiproduit étant donné que l'on a un flot fixé sur certains liens ou sur tous les liens du réseau.

Une application possible serait la suivante :

(a) effectuer un premier calcul du flot maximal multiproduit avec une demande maximale $g_i^{p\,max}$ égale à la quantité que l'on veut envoyer pour chaque paire O/D i et pour chaque produit p. Les flots fixés sur chaque lien sont mis à la valeur zéro.

(b) puis effectuer un deuxième calcul du flot maximal multiproduit avec les flots fixés égaux aux flots calculés en (a). La demande maximale $g_i^{p\,max}$ peut être égale à la même valeur que précédemment. Cela donne une idée de la quantité supplémentaire de produit pouvant être envoyée sur le réseau étant donné qu'une certaine quantité a déjà été affectée sur ce même réseau.

Le nouveau problème s'énonce de la façon suivante :

$$\text{Max} \quad \sum_{p\in P} \sum_{i\in I} g_i^p \gamma_i^p$$

$$\text{sujet à :} \quad \sum_{k\in K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p\in P \text{ et pour toute paire O/D } i\in I$$

$$h_k \geq 0,\ k\in K_i^p \qquad \text{pour tout produit } p\in P \text{ et pour toute paire O/D } i\in I$$

$$\sum_{p\in P} f_a^p + \bar{f}_a \leq Cap_a \qquad a\in A$$

$$f_a^p \geq 0 \qquad a\in A,\ p\in P$$

où $\bar{f}_a$ est le flot fixé sur le lien a (le flot fixé sur chaque lien est 0 s'il n'y a pas de flot fixé)

Il est clair que si l'on donne la valeur zéro aux flots fixés de tous les liens, soit si $f_a = 0$ pour tout $a \varepsilon A$, cela revient à résoudre le problème original donné en (1.1).

1.5.1.2 Modifications de l'algorithme

La fonction du lagrangien augmenté est la suivante :

$$L_{pen}(f_a ; \lambda) = - \sum_{p \varepsilon P} \sum_{i \varepsilon I} g_i^p \gamma_i^p + 1/(2.pen) . \sum_{a \varepsilon A} [\max \{0 ; \lambda_a + pen. (\sum_{p \varepsilon P} f_a^p + \bar{f}_a - Cap_a) \}^2 - \lambda_a^2]$$

avec $pen > 0$ pen est le facteur de pénalité

$\lambda = (\lambda_a)_{a \varepsilon A}$ est le vecteur des multiplicateurs de dimension $| A |$

L'algorithme de résolution proposé par **Bertsekas** subit une modification à l'étape 2 . L'algorithme au complet est :

étape 0 - (Initialisation)

$\ell=0$ (ℓ est le compteur d'itérations)

$\lambda_\ell=0$

étape 1 - (Résolution du sous-problème à coûts convexes)

Calculer $(f_{a,\ell}^p ; g_{i,\ell}^p)$ la solution optimale de **PL** (λ_ℓ)

étape 2 - (Ajustement des multiplicateurs)

$$\lambda_{a,\ell+1} = \max \{ 0 ; \lambda_{a,\ell} + pen_\ell . (\sum_{p \varepsilon P} f_{a,\ell}^p + \bar{f}_a - Cap_a) \} , a \varepsilon A$$

$pen_{\ell+1} \geq pen_\ell$

$\ell=\ell+1$

allera étape 1

Le premier critère d'arrêt sur les multiplicateurs est également modifié de la façon

suivante. On arrête l'algorithme lorsque la norme de la différence entre deux vecteurs de multiplicateurs de deux itérations successives est inférieure à une certaine valeur ε_1 , soit que :

$$\| \lambda_{\ell+1} - \lambda_\ell \| = \| \max \{ - \lambda_\ell ; \text{pen}_\ell . (\sum_{p \in P} f_\ell^p + \bar{f}_\ell - \text{Cap}) \} \| < \varepsilon_1$$

Ces modifications au problème impliquent les modifications suivantes au programme "MAXFLO" donné en annexe B :

(a) On ajoute un autre attribut à chaque lien a qui correspond au flot fixé.
Soit FFLOW le vecteur réel de flot fixé de dimension |A|.

(b) On ajoute dans chaque ligne du fichier de donnée pour chaque lien l'attribut FFLOW.

(c) On modifie les formules de calcul des fonctions SUMTRA,FPRIM et USERCST servant à la résolution du sous-problème par la méthode PARTAN pour y inclure FFLOW. De plus, on modifie le critère d'arrêt sur les multiplicateurs dans le programme principal pour y inclure également le vecteur FFLOW.

Il est facile d'étendre le modèle original pour pouvoir traiter un problème de flot maximal multiproduit avec un flot fixé sur chaque lien : en effet, la contrainte de capacité est modifiée pour l'ajout d'un terme constant, qui implique uniquement un changement dans la fonction du lagrangien augmenté et non dans la structure du programme.

1.5.2 Capacités dépendant du flot dans les deux directions

1.5.2.1 Enoncé du problème

Le problème considéré ici est là encore plus général que celui décrit en (1.1). Il s'agit de calculer le flot maximal multiproduit étant donné que certaines capacités dépendent du flot dans les deux directions.

Soit deux noeuds j_1 et j_2 , un arc a allant de j_1 vers j_2 , et un arc â allant de j_2 vers j_1 . Si la capacité de l'arc a dépend du flot dans les deux directions, alors on a que :

$$f_a + f_{\hat{a}} \leq Cap_a \quad \leftrightarrow \quad f_a + f_{\hat{a}} \leq Cap_{\hat{a}}$$

où f_a est le flot sur l'arc a et $f_{\hat{a}}$ est le flot sur l'arc â.

Cela revient à dire que les arcs a et â ont une capacité conjointe $Cap_a = Cap_{\hat{a}}$.

Cette situation peut arriver dans le réseau pour certains liens, surtout dans le réseau ferroviaire où les trains dans les deux directions utilisent les mêmes rails.

Le nouveau problème s'énonce de la façon suivante :

$$\text{Max} \quad \sum_{p \in P} \sum_{i \in I} g_i^p \gamma_i^p$$

$$\text{sujet à :} \quad \sum_{k \in K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$h_k \geq 0, \; k \in K_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$\sum_{p \in P} f_a^p + \mu^* \left(\sum_{p \in P} f_{\hat{a}}^p \right) \leq Cap_a \qquad a \in A$$

$$f_a^p \geq 0 \qquad a \in A, p \in P$$

où â est le lien dans la direction opposée au lien a
(soit si $a=(j_1,j_2)$ alors $\hat{a}=(j_2,j_1)$)
f_a^p est le flot sur le lien a pour le produit p

$$\mu = \begin{cases} 1 & \text{si le liens a et â ont une capacité conjointe} \\ 0 & \text{sinon} \end{cases}$$

$$Cap_a = \begin{cases} \text{la capacité conjointe de a et â} & \text{si } \mu = 1 \\ \text{la capacité uniquement de a} & \text{si } \mu = 0 \end{cases}$$

Il est clair que si aucun lien n'a de capacité conjointe avec le lien dans la direction opposée, alors cela revient à résoudre le problème original donné en (1.1).

1.5.2.2 Modifications de l'algorithme

La fonction du lagrangien augmenté est la suivante :

$$L_{pen}(f_a ; \lambda) = - \sum_{p \in P} \sum_{i \in I} g_i^p \gamma_i^p + 1/(2.pen) . \sum_{a \in A} \max \{0; \lambda_a + pen . (\sum_{p \in P} f_a^p + \mu^* (\sum_{p \in P} f_{\hat{a}}^p) - Cap_a) \}^2 - \lambda_a^2]$$

avec pen > 0 pen est le facteur de pénalité

$\lambda = (\lambda_a)_{a \in A}$ est le vecteur des multiplicateurs de dimension | A |

L'algorithme de résolution proposé par **Bertsekas** subit une modification à l'étape 2 . L'algorithme au complet est :

étape 0 - (Initialisation)

$\ell=0$ (ℓ est le compteur d'itérations)

$\lambda_\ell=0$

étape 1 - (Résolution du sous-problème à coûts convexes)

Calculer $(f_{a,\ell}^p ; g_{i,\ell}^p)$ la solution optimale de **PL** (λ_ℓ)

étape 2 - (Ajustement des multiplicateurs)

$$\lambda_{a,\ell+1} = \lambda_{\hat{a},\ell+1} = \max\{0; \lambda_{a,\ell} + pen_\ell .(\sum_{p \in P} f_{a,\ell}^p + \mu^* (\sum_{p \in P} f_{\hat{a},\ell}^p) - Cap_a)\} , a \in A$$

$pen_{\ell+1} \geq pen_\ell$

$\ell=\ell+1$

allera étape 1

Le premier critère d'arrêt sur les multiplicateurs est également modifié de la façon suivante. On arrête l'algorithme lorsque la norme de la différence entre deux vecteurs de multiplicateurs de deux itérations successives est inférieur à une certaine valeur ε_1, soit que :

$$\|\lambda_{\ell+1} - \lambda_\ell\| = \| \max \{ -\lambda_\ell ; \text{pen}_\ell . (\sum_{p \in P} f_\ell^p + \mu * (\sum_{p \in P} \overrightarrow{f_\ell^p}) - \text{Cap}) \} \| < \varepsilon_1$$

où $\overrightarrow{f_\ell^p}$ est le vecteur de flot des liens dans la direction opposée pour le produit p.

Ces modifications au problème impliquent les modifications suivantes au programme "MAXFLO" donné en annexe B :

(a) On indique par une capacité négative les liens qui ont une capacité conjointe avec le lien dans la direction opposée.

(b) On modifie les formules de calcul des fonctions SUMTRA,FPRIM et USERCST servant à la résolution du sous-problème par la méthode PARTAN pour y inclure le terme: $\mu * (\sum_{p \in P} f_{\hat{a},\ell}^p)$.

De plus, on modifie le critère d'arrêt sur les multiplicateurs dans le programme principal pour y inclure aussi le terme $\mu * (\sum_{p \in P} f_{\hat{a},\ell}^p)$.

Il est facile d'étendre le modèle original pour pouvoir traiter un problème de flot maximal multiproduit avec des capacités dépendant du flot dans les deux directions : en effet, la contrainte de capacité est modifiée par l'ajout du terme $\mu * (\sum_{p \in P} f_{\hat{a}}^p)$, qui implique uniquement un changement dans la fonction du lagrangien augmenté et non dans la structure du programme.

1.5.3 Capacités différentes par combinaison de produit

Tous les produits n'utilisent pas les mêmes ressources d'emmagasinage ni de transport Tout particulièrement dans les ports, les différents produits sont emmagasinés dans différents endroits : par exemple, les grains par opposition aux autres produits sont placés dans des silos, et non dans des entrepôts. Cela implique que dans le réseau considéré, pour calculer la quantité maximale de produit pouvant être envoyée de chaque origine vers chaque destination, il n'est pas suffisant d'avoir une capacité par lien car cette capacité englobe tous les produits.

1.5.3.1 Représentation du réseau

Pour pouvoir avoir différentes capacités pour différentes combinaisons de produits, il est nécessaire d'apporter des modifications aux structures de données utilisées par le programme "MAXFLO" et/ou à la représentation du réseau. Les structures de données actuellement utilisées pour représenter le réseau sont telles que l'on a une seule capacité par lien pour l'ensemble des produits. On pourrait avoir une structure de données qui permettrait des capacités différentes par combinaison de produit pour chaque lien du réseau mais cela demanderait beaucoup d'espace en mémoire pour emmagasiner les différentes capacités et surtout cela obligerait à garder le flot de chaque lien a par produit ou combinaison de produit au lieu de garder en mémoire uniquement la somme des flots de tous les produits sur le lien a, soit $\sum_{p \in P} f_a^p$.

Donc il est préférable de garder une seule capacité par lien mais de ne permettre que certains produits sur les liens qui ont une capacité différente par combinaison de produit : les produits permis sont ceux ayant une capacité conjointe. Cela implique une modification de la structure du réseau pour les liens considérés, soit une augmentation du nombre de liens et du nombre de noeuds dans le réseau.

Par exemple, si sur un certain lien allant de noeud j_1 au noeud j_2, les produits 1 et 3 ont une capacité conjointe, les produits 2,5 et 6 ont une capacité conjointe, le produit 4 a une certaine capacité non partagée avec les autres produits, et que le nombre total de produits est 6, alors la nouvelle représentation est donnée à la figure (1.1). On remarque que trois noeuds et six liens ont été ajoutés. Ils correspondent aux trois combinaisons de produits avec des capacités différentes.

Les modifications apportées à la représentation et aux structures de données du programme "MAXFLO" sont les suivantes :

(a) Modification de la représentation du réseau :

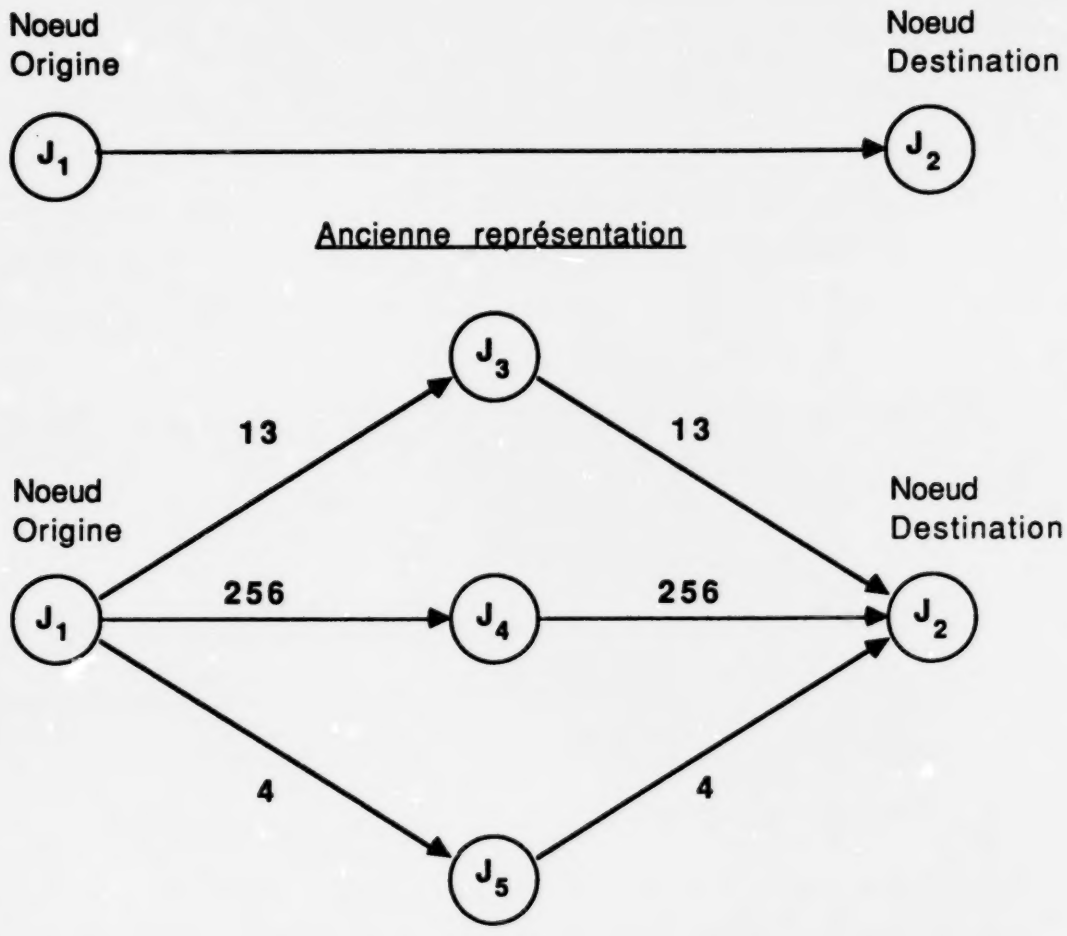
Pour les liens qui possèdent une capacité différente par ß combinaisons de produit, on ajoute β noeuds intermédiaires entre le noeud origine et le noeud destination (pour ne pas avoir de liens parallèles) et (2*β) liens reliant le noeud origine, le noeud destination et les β noeuds intermédiaires. Avec cette représentation, on garde une seule capacité par lien.

(b) Modification des structures de données du programme 'MAXFLO' :

On doit ajouter un attribut par lien indiquant la combinaison de produits permise sur ce lien.

1.5.3.2 Produits permis sur un lien

Comme mentionné plus haut, nous devons avoir pour chaque lien une information supplémentaire indiquant les produits ou combinaisons de produit permis sur chaque lien. Un autre attribut a donc été ajouté à chaque lien du réseau indiquant les produits ou combinaisons de produit permis sur chaque lien. Dans le programme "MAXFLO", cet attribut



Légende :

1...6 : code pour les six produits
J_1 ... J_5 : code pour les noeuds

Figure 1.1 : Ancienne et nouvelle représentation du réseau pour considérer les capacités différentes par combinaisons de produits

est le vecteur d'entiers IPROD de dimension |A|.

Pour |P| produits, nous avons $\sum_{i=1}^{|P|} C_i^{|P|}$ combinaisons possibles de produits.
Pour six produits, par exemple, cela nous donne $\sum_{i=1}^{6} C_i^{6}=63$ combinaisons possibles de produits

On peut établir une table de correspondance entre iPROD et les produits permis sur le lien. Le tableau (1.1) donne la correspondance entre la variable IPROD et les produits permis sur un lien dans le cas où le nombre de produits |P|=6.

1.5.3.3 Modifications de l'algorithme

La fonction du lagrangien augmenté ainsi que l'algorithme de résolution du problème sont les mêmes que pour le problème original.

Les modifications apportées au problème sont les modifications spécifiées plus haut, soit :

(a) On modifie la représentation du réseau par l'ajout de noeuds (pour ne pas avoir de liens parallèles) et de liens (pour avoir une capacité par lien).

(b) On ajoute un attribut par lien indiquant la combinaison de produits permise sur ce lien. IPROD contient cette information. On ajoute par conséquent dans chaque ligne du fichier de donnée pour chaque lien l'attribut IPROD.

De plus, on effectue les modifications suivantes :

(c) On ajoute une procédure IPERM permettant de déterminer si un produit p est permis sur l'arc a étant donné l'attribut donné IPROD. Cette fonction est utilisée dans tous les calculs impliquant la fonction lagrangienne, soit dans les fonctions SUMTRA, FPRIM et USERCST dans les calculs des critères d'arrêt et dans l'ajustement des multiplicateurs.

IPROD	PRODUITS PERMIS SUR LE LIEN
0	1-2-3-4-5-6
1	1
2	2
3	3
4	4
5	5
6	6
12	1-2
13	1-3
14	1-4
15	1-5
16	1-6
23	2-3
24	2-4
25	2-5
26	2-6
34	3-4
35	3-5
36	3-6
45	4-5
46	4-6
56	5-6
123	1-2-3
124	1-2-4
125	1-2-5
126	1-2-6
134	1-3-4
135	1-3-5
136	1-3-6
145	1-4-5
146	1-4-6
156	1-5-6

IPROD	PRODUITS PERMIS SUR LE LIEN
-1	2-3-4-5-6
-2	1-3-4-5-6
-3	1-2-4-5-6
-4	1-2-3-5-6
-5	1-2-3-4-6
-6	1-2-3-4-5
-12	3-4-5-6
-13	2-4-5-6
-14	2-3-5-6
-15	2-3-4-6
-16	2-3-4-5
-23	1-4-5-6
-24	1-3-5-6
-25	1-3-4-6
-26	1-3-4-5
-34	1-2-5-6
-35	1-2-4-6
-36	1-2-4-5
-45	1-2-3-6
-46	1-2-3-5
-56	1-2-3-4
234	2-3-4
235	2-3-5
236	2-3-6
245	2-4-5
246	2-4-6
256	2-5-6
345	3-4-5
346	3-4-6
356	3-5-6
456	4-5-6

Tableau 1.1 : Tableau de correspondance entre la variable IPROD et les produits permis sur le lien

L'extension du modèle original pour pouvoir traiter un problème de flot maximal multiproduit avec des capacités différentes par combinaison de produit implique les deux modifications principales suivantes : on modifie la représentation du réseau pour garder la structure de une capacité par lien et on ajoute un attribut pour indiquer les produits permis sur le lien.

Cette extension du modèle sera particulièrement utile dans les ports où les produits n'utilisent pas le même matériel de transport ni d'entreposage. Remarquons que le nombre de liens et de noeuds à ajouter est minime si l'on considère le réseau dans sa totalité.

1.5.4 Borne inférieure sur la demande de certaines paires O/D pour certains produits

1.5.4.1 Enoncé du problème

Pour certaines paires O/D et pour certains produits, le planificateur peut vouloir forcer la quantité de ces produits pour ces paires O/D à avoir au moins une certaine valeur. Donc, pour ces paires O/D $i \in I$ et pour ces produits $p \in P$, la solution du problème, soit la quantité maximale de produit p pour ces paires O/D i g_i^p doit être plus grande ou égale à une demande minimale $g_i^{p\,min}$. Cette décision du planificateur peut être due à une politique de contraindre une certaine quantité d'un produit particulier à être envoyée sur le réseau. Cela implique l'ajout de contraintes additionnelles au problème original.
Le nouveau problème peut donc être formulé de la façon suivante :

$$\text{Max} \quad \sum_{p \in P} \sum_{i \in I} g_i^p \gamma_i^p$$

$$\text{sujet à :} \quad \sum_{k \in K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$h_k \geq 0,\ k \in K_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$\sum_{p \in P} f_a^p \leq Cap_a \quad a \in A$$

$$f_a^p \geq 0 \qquad a \in A, p \in P$$

$$g_i^p \geq g_i^{p\,min} \qquad i \in I, p \in P$$

où $g_i^{p\,min}$ est la borne inférieure sur la demande de la paire O/D i pour le produit p.

Il est clair que si aucune demande n'est contrainte à être supérieure à une borne inférieure $g_i^{p\,min}$ (c'est-à-dire $g_i^{p\,min} = 0$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$) alors cela revient à résoudre le problème original.

Remarquons également que le problème peut n'avoir aucune solution réalisable si les contraintes ajoutées sur les g_i^p ne permettent pas de trouver des flots sur les liens $\sum_{p \in P} f_a^p$ qui sont inférieurs à la capacité du lien Cap_a.

On propose deux méthodes de résolution basées sur le lagrangien augmenté :

(a) la première méthode est une approche lagrangienne augmentée pour les contraintes ajoutées. Les contraintes ajoutées sont relaxées et un multiplicateur est associé à chacune d'elle.

(b) la deuxième méthode est une modification directe de l'algorithme utilisé pour résoudre le sous-problème PL(λ). Les contraintes ajoutées ne sont pas relaxées.

Les deux méthodes sont présentées et comparées dans les prochaines sections. Des résultats numériques concernant ces deux méthodes sont également exposés dans la section (1.6.9).

1.5.4.2 Première méthode de résolution : approche lagrangienne augmentée

La première méthode utilisée pour résoudre ce problème est la méthode du **lagrangien augmenté** où les contraintes ajoutées sur la demande pour certains produits $p \in P$ pour certaines paires O/D $i \in I$ sont traitées de la même façon que les contraintes de capacité. On associe un multiplicateur λ_i^p, $i \in I$ et $p \in P$ à chacune de ces contraintes et on définit une nouvelle fonction du lagrangien augmenté tenant compte de ces contraintes.

La fonction du lagrangien augmenté est :

$$L_{pen}(f_a ; \lambda) = - \sum_{p \in P} \sum_{i \in I} g_i^p \gamma_i^p + 1/(2.pen) \cdot \sum_{a \in A} [\max \{0; \lambda_a + pen \cdot (\sum f_a^p - Cap_a)\}^2 - \lambda_a^2] + 1/(2.pen) \cdot \sum_{(i,p) \in IP} [\max \{0; \lambda_i^p + pen \cdot (g_i^{p\,min} - g_i^p)\}^2 - \lambda_i^{p\,2}]$$

avec $pen > 0$ pen est le facteur de pénalité

$\lambda = (\lambda_a \; a \in A ; \lambda_i^p \; i \in I, p \in P)$ est le vecteur des multiplicateurs de dimension $|A| + |IP|$

IP est l'ensemble des indices (i,p) des paires O/D $i \in I$ et des produits $p \in P$ pour lesquels on a une borne inférieure $g_i^{p\,min}$.

On peut encore écrire le modèle de la façon suivante :

PL (λ) $L_{pen}(\lambda) = \text{Min } L_{pen}(f_a ; \lambda)$

sujet à : $\sum_{k \in K_i^p} h_k = g_i^p$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$

$h_k \geq 0, \; k \in K_i^p$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$

et nous avons que

PL $L^* = \underset{\lambda \geq 0}{\text{Max}} \; L_{pen}(\lambda)$

L'algorithme pour résoudre le problème **PL** subit une modification à l'étape 2.

L'algorithme de résolution est le suivant :

étape 0 - (Initialisation)

$\ell=0$ (ℓ est le compteur d'itérations)

$\lambda_\ell=0$

étape 1 - (Résolution du sous-problème à coûts convexes)

Calculer $(f_{a,\ell}^p ; g_{i,\ell}^p)$ la solution optimale de **PL** (λ_ℓ)

étape 2 - (Ajustement des multiplicateurs)

$$\lambda_{a,\ell+1} = \max\{0 ; \lambda_{a,\ell} + pen_\ell . (\sum_{p\in P} f_{a,\ell}^p - Cap_a)\}, a\in A$$

$$\lambda_{i,\ell+1}^p = \max\{0 ; \lambda_{i,\ell}^p + pen_\ell . (g_i^{p\,min} - g_{i,\ell}^p)\}, i\in I, p\in P$$

$pen_{\ell+1} \geq pen_\ell$; $\ell = \ell+1$; allera étape 1

On modifie les critères d'arrêt en conséquence pour tenir compte des multiplicateurs ajoutés pour les paires O/D i et les produits p qui interviennent dans les contraintes ajoutées.

Le premier critère d'arrêt de l'algorithme concernant les multiplicateurs devient le suivant. On arrête l'algorithme lorsque la norme de la différence entre deux vecteurs multiplicateurs de deux itérations successives est inférieure à une certaine valeur ε_1, soit que :

$$\| \lambda_{\ell+1} - \lambda_\ell \| < \varepsilon_1$$

où $\| \lambda_{\ell+1} - \lambda_\ell \| = \| \max\{ \max_{a\in A}\{ -\lambda_{a,\ell} ; pen_\ell . (\sum_{p\in P} f_{a,\ell}^p - Cap_a)\} ;$

$$\max_{i\in I, p\in P}\{ -\lambda_{i,\ell}^p ; pen_\ell . (g_i^{p\,min} - g_{i,\ell}^p)\}\} \| < \varepsilon_1$$

Explicitons maintenant la méthode de résolution du sous-problème de l'étape 1 de l'algorithme énoncé ci-haut. La méthode de résolution est la même que pour le problème original.

Le sous-problème PL(λ) de l'étape 1 de l'algorithme de résolution peut-être vu comme un problème de multiflot d'équilibre avec demande variable.
Le sous-problème est de la forme :

$$PL(\lambda)\ \text{Min} \sum_{a\varepsilon A} s_a(f_a) - \sum_{p\varepsilon P} \sum_{i\varepsilon I} \int_0^{g_i^p} [\gamma_i^p + \max\{0; \lambda_i^p + pen.(g_i^p \min - g_i^p)\}] \cdot \partial g_i^p$$

$$\text{sujet à :} \sum_{k\varepsilon K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p\varepsilon P \text{ et pour toute paire O/D } i\varepsilon I$$

$$h_k \geq 0,\ k\varepsilon K_i^p \qquad \text{pour tout produit } p\varepsilon P \text{ et pour toute paire O/D } i\varepsilon I$$

où $s_a(f_a)$ est la partie de la fonction relaxée pour les contraintes de capacité

$$s_a(f_a) = 1/(2.pen) \cdot [\max\{0; \lambda_a + pen_{\lambda} \cdot (\sum_{p\varepsilon P} f_a^p - Cap_a)\}^2 - \lambda_a^2]$$

f_a est le vecteur des flots.

Pour résoudre ce type de problème, on utilise comme pour le problème original une variante de l'algorithme Frank & Wolfe : la méthode Partan. On fait des développements similaires à ceux effectués en (1.4.2) pour calculer la solution optimale du sous-problème linéaire du problème $PL(\lambda_{\lambda})$ pour λ_{λ} fixé.

Soit, comme à la section (1.4.2),

la variable $H_k,\ k\varepsilon K_i^p$ correspond à la variable $h_k,\ k\varepsilon K_i^p$

la variable w_i^p correspond à la variable g_i^p

la variable F_a^p correspond à la variable f_a^p.

Le sous-problème linéaire associé au problème PL (λ_2) à résoudre est le suivant :

$$\text{Min} \sum_{p\in P} \sum_{i\in I} \sum_{k\in K_i^p} \left(\sum_{a\in A} \delta_{ak} \cdot \partial s_a(f_a) / \partial h_k \right) H_k$$
$$- \sum_{p\in P} \sum_{i\in I} (\gamma_i^p + \max \{ o; \lambda_i^p + pen \cdot (g_i^{p\,min} - g_i^p) \}) \cdot w_i^p$$

sujet à : $\sum_{k\in K_i^p} H_k = w_i^p$ pour tout produit $p\in P$ et pour toute paire O/D $i\in I$

$H_k \geq 0,\ k\in K_i^p$ pour tout produit $p\in P$ et pour toute paire O/D $i\in I$

La fonction objective devient à l'itération r :

$$\sum_{p\in P} \sum_{i\in I} \left(\sum_{k\in K_i^p} s'_{kr}\, H_k - [\gamma_i^p + \max \{ o; \lambda_i^p + pen \cdot (g_i^{p\,min} - g_i^p) \}] \cdot w_i^p \right)$$

où $s'_{kr} = \sum_{a\in A} \delta_{ak} \cdot \partial s_a(f_a) / \partial h_k$ à l'itération r

s'_{kr} est le "coût" sur le chemin k à l'itération r.

$$= \sum_{p\in P} \sum_{i\in I} \left(\sum_{k\in K_i^p} u_i H_k - [\gamma_i^p + \max \{ o; \lambda_i^p + pen \cdot (g_i^{p\,min} - g_i^p) \}] \cdot w_i^p \right)$$

où u_i est le coût du plus court chemin pour la paire O/D $i\in I$

$$= \sum_{p\in P} \sum_{i\in I} \left(u_i \sum_{k\in K_i^p} H_k - [\gamma_i^p + \max \{ o; \lambda_i^p + pen \cdot (g_i^{p\,min} - g_i^p) \}] \cdot w_i^p \right)$$

$$= \sum_{p\in P} \sum_{i\in I} \left(u_i - [\gamma_i^p + \max \{ o; \lambda_i^p + pen \cdot (g_i^{p\,min} - g_i^p) \}] \right) \cdot w_i^p$$

à cause de la contrainte $\sum_{k\in K_i^p} H_k = w_i^p$

Le problème de l'étape 1 de l'algorithme PARTAN revient donc à :

$$\text{Min} \sum_{p\in P} \sum_{i\in I} \left(u_i - [\gamma_i^p + \max \{ o; \lambda_i^p + pen \cdot (g_i^{p\,min} - g_i^p) \}] \right) \cdot w_i^p$$

sujet à : $w_i^p \geq 0,\quad i\in I,\ p\in P$

Le sous-problème de PL (λ) se résoud facilement puisque :

1) si $u_i < [\gamma_i^p + \max \{ o; \lambda_i^p + pen \cdot (g_i^{p\,min} - g_i^p) \}]$, nous avons intérêt à maximiser w_i^p

2) si $u_i > [\gamma_i^p + \max\{0; \lambda_i^p + pen.(g_i^{p\,min} - g_i^p)\}]$, nous avons intérêt à minimiser w_i^p.

La solution de ce sous-problème est obtenue en calculant les chemins les plus courts u_i pour chaque paire O/D $i \in I$. Puis les demandes w_i^p sont obtenues de cette façon : pour chaque paire O/D $i \in I$ et pour chaque produit $p \in P$,

$$w_i^p = \begin{cases} g_i^{p\,max} & \text{si } u_i < \gamma_i^p + \max\{0; \lambda_i^p + pen.(g_i^{p\,min} - g_i^p)\} \\ g_{i,r}^p & \text{si } |u_i - \gamma_i^p - \max\{0; \lambda_i^p + pen.(g_i^{p\,min} - g_i^p)\}| < \varepsilon \\ 0 & \text{si } u_i > \gamma_i^p + \max\{0; \lambda_i^p + pen.(g_i^{p\,min} - g_i^p)\} \end{cases} \quad (1.1)$$

La somme des flots sur un arc pour tous les produits $\sum_{p \in P} f_a^p$ est obtenue par conséquent en affectant $\sum_{p \in P} w_i^p$ sur le plus court chemin pour toute paire O/D i.

La direction de descente est : $(d_r ; e_r) = (F_{a,r}^p ; w_{i,r}^p) - (f_{a,r}^p ; g_{i,r}^p)$

où $F_{a,r}^p = \sum_{i \in I} \sum_{k \in K_i^p} \delta_{ak} H_k$ et $\delta_{ak} = \begin{cases} 1 & \text{si l'arc a appartient au chemin k} \\ 0 & \text{sinon} \end{cases}$

Donc le seul changement dans l'algorithme de résolution de **PL (λ)** est le calcul de w_i^p pour la direction de descente Frank & Wolfe de l'étape 1. Les w_i^p sont donnés par la formule **(1.1)**. Cette formule inclut le terme relaxé correspondant aux contraintes ajoutées.

Les modifications à apporter à l'algorithme original pour l'implantation de cette première méthode de résolution sont :

(a) On ajoute à chaque ligne de donnée des paires O/D l'information concernant la demande minimale $g_i^{p\,min}$: cette information est emmagasinée dans une matrice GMIN.

(b) On ajoute des multiplicateurs pour chaque contrainte additionnelle.On utilise une matrice pour mémoriser ces multiplicateurs. On modifie le calcul du critère d'arrêt sur les multiplicateurs pour tenir compte des multiplicateurs ajoutés.

(c) On modifie la procédure TRAFFIC pour le calcul de la direction de descente.

Les w_i^p sont donnés par la formule **(1.1)**.

1.5.4.3 Deuxième méthode de résolution : modification directe de l'algorithme de résolution

La deuxième méthode utilisée pour résoudre ce problème est une modification directe de l'algorithme utilisé pour résoudre le sous-problème PL(λ). On ne relaxe pas les contraintes sur g_i^p. La méthode de base utilisée est, comme au problème original de la section **(1.1)**, la méthode du **lagrangien augmenté**. La fonction du lagrangien augmenté est la même que pour le problème original de (1.1). On n'a qu'un seul vecteur de multiplicateurs pour les liens du réseau : $\lambda=(\lambda_a)_{a\in A}$. Ce vecteur de multiplicateurs est associé aux contraintes de capacités. Contrairement à la première méthode de résolution, on n'utilise pas des multiplicateurs λ_i^p, $i\in I$, $p\in P$ pour chaque nouvelle contrainte ajoutée.

On peut écrire le modèle de la façon suivante :

PL (λ) $L_{pen}(\lambda) = \text{Min } L_{pen}(f_a, \lambda)$

sujet à : $\sum_{k\in K_i^p} h_k = g_i^p$ pour tout produit $p\in P$ et pour toute paire O/D $i\in I$

$h_k \geq 0,\ k\in K_i^p$ pour tout produit $p\in P$ et pour toute paire O/D $i\in I$

$g_i^p \geq g_i^{p\,min}$ $(I,p) \in IP$

et nous avons que

$$\text{PL} \qquad L^* = \max_{\lambda \geq 0} L_{pen}(\lambda)$$

L'algorithme de résolution du problème ne subit donc aucune modification, ni les critères d'arrêt.

Explicitons maintenant la méthode de résolution du sous-problème de l'étape 1 de l'algorithme de résolution. La méthode de résolution est la même que pour le problème original, sauf qu'il faut tenir compte des contraintes additionnelles sur la demande prédite :

$$g_i^p \geq g_i^p \text{ min} \qquad (i,p) \in IP$$

Le sous-problème PL(λ) de l'étape 1 de l'algorithme peut encore être vu comme un problème de multiflot d'équilibre avec demande variable. Le sous-problème est le même qu'à la section (1.4) mais avec les contraintes additionnelles sur la demande g_i^p, soit :

$$\text{PL}(\lambda) \quad \text{Min} \sum_{a \in A} s_a(f_a) - \sum_{p \in P} \sum_{i \in I} \int_0^{g_i^p} \gamma_i^p \, \partial g_i^p$$

$$\text{sujet à :} \quad \sum_{k \in K_i^p} h_k = g_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$h_k \geq 0, \; k \in K_i^p \qquad \text{pour tout produit } p \in P \text{ et pour toute paire O/D } i \in I$$

$$g_i^p \geq g_i^p \text{ min} \qquad (i,p) \in IP$$

où $s_a(f_a)$ est la partie de la fonction relaxée pour les contraintes de capacité

$$s_a(f_a) = 1/(2.pen) \cdot [\max\{0; \lambda_a + pen_\lambda \cdot (\sum_{p \in P} f_a^p - Cap_a)\}^2 - \lambda_a^2]$$

f_a est le vecteur des flots.

La méthode PARTAN est utilisée pour résoudre le sous-problème PL(λ_λ).

Le sous-problème linéaire associé au problème PL(λ_λ) à résoudre est le même qu'en (1.4.2) avec les contraintes additionnelles sur la demande, soit :

$$\text{Min} \sum_{p \in P} \sum_{i \in I} \sum_{k \in K_i^p} \left(\sum_{a \in A} \delta_{ak} \cdot \partial s_a(f_a) / \partial h_k \right) H_k - \sum_{p \in P} \sum_{i \in I} \gamma_i^p \cdot w_i^p$$

sujet à : $\sum_{k \in K_i^p} H_k = w_i^p$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$

$H_k \geq 0,\ k \in K_i^p$ pour tout produit $p \in P$ et pour toute paire O/D $i \in I$

$g_i^p \geq g_i^p \text{ min}$ $(I,p) \in IP$

On fait les mêmes transformations qu'en (1.4.2) sur la fonction objective. Le sous-problème linéaire associé au problème $PL(\lambda_\ell)$ revient donc à :

$$\text{Min} \sum_{p \in P} \sum_{i \in I} (u_i - \gamma_i^p) \cdot w_i^p$$

sujet à : $w_i^p \geq g_i^p \text{ min}, \quad (i,p) \in IP$

$w_i^p \geq 0 \quad , \quad i \in I,\ p \in P$

où u_i est le coût du plus court chemin pour la paire O/D $i \in I$.

Ce qu'il y a à modifier directement dans la méthode Partan est la détermination des w_i^p pour la direction de descente. La solution de ce sous-problème est obtenue en calculant les chemins les plus courts u_i pour chaque paire O/D $i \in I$. Puis on obtient les demandes w_i^p de la façon suivante : pour chaque paire O/D $i \in I$ et pour chaque produit $p \in P$,

$$w_i^p = \begin{cases} g_i^p \text{ max} & \text{si } u_i < \gamma_i^p \\ g_{i,r}^p & \text{si } |u_i - \gamma_i^p| < \varepsilon \\ g_i^p \text{ min} & \text{si } u_i > \gamma_i^p \end{cases} \qquad (1.2)$$

La somme des flots sur un arc pour tous les produits $\sum_{p \in P} f_a^p$ est obtenue par conséquent en affectant $\sum_{p \in P} w_i^p$ sur le plus court chemin pour toute paire O/D i.

Comme il a été mentionné plus haut à la section (1.5.4.1), il est probable que la solution ne soit pas **réalisable** à cause des contraintes de capacité. En effet, en affectant

$\sum_{p \in P} w_i^p$ sur le plus court chemin pour toute paire O/D $i \in I$, il est possible que l'on obtienne que : $\sum_{p \in P} f_a^p > Cap_a$ pour certains liens a puisque $w_i^p \geq g_i^{p\,min}$ pour $(i,p) \in IP$.

Les modifications à apporter à l'algorithme original pour cette deuxième méthode de résolution sont minimes comparées à celle de la première méthode. Les modifications sont les suivantes :

(a) On ajoute à chaque ligne de donnée concernant les paires O/D l'information concernant la demande minimale $g_i^{p\,min}$: cette information est emmagasinée dans une matrice GMIN.

(b) On modifie la procédure TRAFFIC pour le calcul de la direction de descente.

Dans le cas où le coût du plus court chemin u_i est supérieur au poids de la paire O/D i pour le produit p γ_i^p, alors on affecte à w_i^p la valeur $g_i^{p\,min}$ et non la valeur zéro.

Si l'on compare les deux méthodes, on remarque que la deuxième méthode requiert beaucoup moins d'espace en mémoire que la première méthode étant donné que la deuxième méthode ne nécessite pas la mémorisation d'un vecteur de multiplicateurs pour les contraintes supplémentaires. De plus, les modifications et les calculs additionnels sont moindres pour la deuxième méthode : en effet, la fonction du lagrangien augmenté, le critère d'arrêt sur les multiplicateurs, la détermination de la direction de descente dans la résolution du sous-problème $PL(\lambda_k)$ sont plus complexes pour la première méthode, à cause des contraintes additionnelles.

Il semble donc à priori que la deuxième méthode soit plus efficace que la première méthode. Des tests effectués sur des réseaux de petite taille sont présentés à la section (1.6.9) pour comparer en pratique les deux méthodes.

1.6 Résultats numériques

La méthode de résolution du problème de calcul du flot maximal multiproduit ainsi que les méthodes de résolution des extensions du modèle ont été programmées et testées sur des réseaux de petite taille.

Les tests portant sur le modèle original montrent l'influence sur la solution optimale de la variation du facteur de pénalité PEN, de la variation du poids de chaque paire O/D i pour chaque produit p, et de la variation de la demande maximale $g_i^{p\,max}$.

Parmi les réseaux de petite taille testés, on a utilisé un réseau de 13 noeuds, 4 centroïdes, 19 liens baptisé le réseau "Dupuis" (Nguyen et Dupuis [1981]).

Le programme "MAXFLO" avec lequel les tests ont été effectués a été écrit en FORTRAN 4 et consiste en une modification du programme "TRAFFIC" (Nguyen et James [1975]) et du programme "TRAFVAR" (André Babin [1980]). Le code est fourni en annexe B. L'ordinateur utilisé pour réaliser les divers tests est le CDC CYBER 835 de l'Université de Montréal.

1.6.1 Réseau "Dupuis"

Ce réseau imaginaire est tiré de la publication de S. Nguyen et C. Dupuis (Nguyen et Dupuis [1981]). Ce réseau comprend :13 noeuds, 4 centroïdes, 19 liens et 4 paires Origine/Destination. Le réseau "Dupuis" est donné à la figure (1.2).

1.6.2 Choix des paramètres

Nous tenterons dans cette sous-section d'identifier les différents choix qu'il a fallu effectuer en rapport avec certains paramètres d'ajustement. Il faut bien se rendre compte

Le réseau "Dupuis" est le suivant :

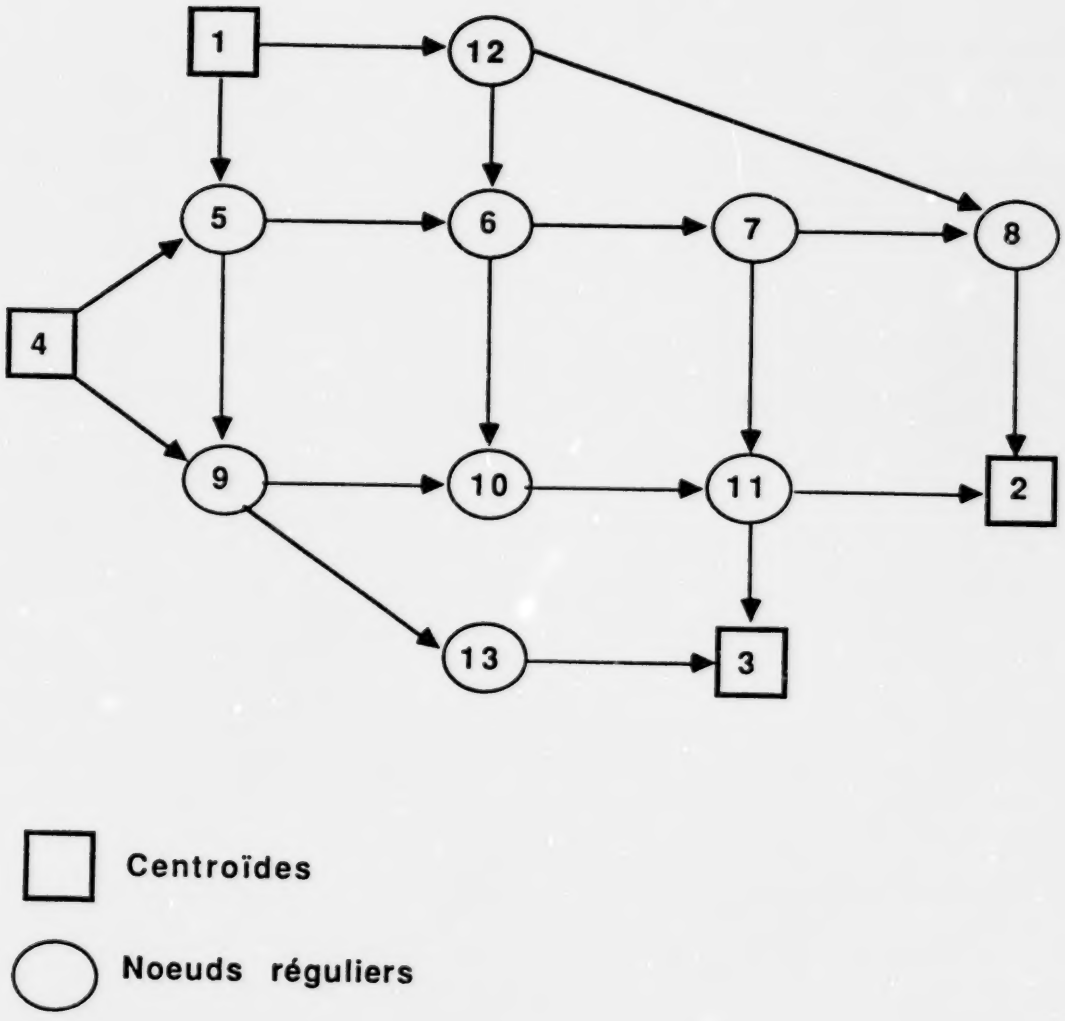


Figure 1.2 : Réseau "Dupuis"

ici que ces choix sont intuitifs : ils constituent donc un ensemble de paramètres d'ajustement. Aussi une même méthode de base peut engendrer différentes solutions, plus ou moins bonnes les unes par rapport aux autres, et ce en fonction du même problème de base. Ainsi pour un problème particulier donné, seule la génération de plusieurs solutions issues de différents ensembles de choix de paramètres d'ajustement peut nous permettre de connaître les meilleurs choix à effectuer. Pour l'instant nous nous contentons d'effectuer les choix qui nous semblent les plus logiques, afin de mettre au point une méthodologie de résolution. Dans une deuxième phase, nous pourrons nous attarder à l'amélioration de la méthode.

Dans la méthode de résolution, on a un très grand nombre de paramètres auxquels il faut donner une valeur. Les paramètres peuvent être divisés en deux groupes : ceux au niveau de la résolution du problème global et ceux au niveau de la résolution du sous-problème PL(λ) où le vecteur de multiplicateurs λ est fixé.

Au niveau du problème global PL, les paramètres sont les suivants :

- le nombre maximum d'itérations
- le paramètre de précision ε_1 du critère d'arrêt sur les multiplicateurs : ce paramètre est mis à une valeur pas trop grande (par exemple : 0.05) car les multiplicateurs varient beaucoup d'une itération à l'autre
- le paramètre de précision ε_2 du critère d'arrêt sur le flot prédit
- le paramètre de précision ε_3 du critère d'arrêt sur la demande prédite (c'est-à-dire sur la quantité de produit p pour la paire O/D i prédite)
- le poids de chaque paire O/D i pour chaque produit p : ce paramètre influe beaucoup sur la solution car il permet de donner plus d'importance à une certaine paire O/D i pour un certain produit p
- le paramètre de pénalité PEN : comme mentionné à la section (1.3), le choix de PEN influe sur le conditionnement et sur la convergence du problème primal et du problème dual.

Dans toute méthode avec fonction de pénalité, la séquence de paramètres PEN est choisie à

l'avance : PEN est soit maintenu fixé, soit augmenté jusqu'à une certaine valeur finie.

Au niveau du sous-problème PL(λ) où λ est fixé, les paramètres sont les suivants :

- le nombre maximum d'itérations
- le paramètre de précision ε du critère d'arrêt de l'algorithme de type Partan.
- le paramètre de précision de la recherche unidimensionnelle de l'algorithme de type Partan
- la valeur maximale de la demande prédite pour chaque paire O/D i de chaque produit p $g_i^{p\,max}$. Ce paramètre influe lui-aussi beaucoup la solution optimale par le biais de la direction de descente.

Nous allons faire une analyse approfondie de l'influence sur la solution optimale des trois paramètres suivants :

- le paramètre de pénalité PEN
- le poids de chaque paire O/D pour chaque produit p : γ_i^p
- la valeur maximale de la demande pour chaque paire O/D i de chaque produit p : $g_i^{p\,max}$

1.6.3 Variation du facteur de pénalité PEN

Des tests ont été effectués sur le réseau "Dupuis" avec les caractéristiques suivantes :

- le nombre de produit est 1
- la capacité de chaque lien est 40.0

Les paramètres choisis pour les tests sur le réseau "Dupuis" sont les suivants:

Au niveau du problème global PL,

- Nombre maximum d'itérations : 20
- Paramètre de précision ε_1 du critère d'arrêt sur les multiplicateurs : 0.05
- Paramètre de précision ε_2 du critère d'arrêt sur le flot prédit : 0.00005

- Paramètre de précision ε_3 du critère d'arrêt sur la demande prédite : 0.00005
- Poids de chaque paire O/D i pour chaque produit p : 1.0

Au niveau du sous-problème PL(λ) où λ est fixé,

- Nombre maximum d'itérations : 20
- Paramètre de précision ε du critère d'arrêt : 0.00001
- Valeur maximale de la demande prédite pour chaque paire O/D i de chaque produit p $g_i^{p\,max}$: 100.0
- Paramètre de précision pour la recherche unidimensionnelle : 0.001

On fait varier le facteur de pénalité PEN. Les résultats sont donnés au tableau (1.2). La première remarque que l'on peut faire quant à la solution g_i est que dans les tests effectués, le nombre global d'itérations est moins élevé pour un paramètre de pénalité PEN plus élevé cela correspond bien au fait que pour un paramètre de pénalité PEN très grand la convergence duale est rapide. De plus, la norme de la différence entre les multiplicateurs de deux itérations successives est plus petite pour un paramètre de pénalité PEN plus élevé. Finalement, dans certains cas testés, il s'est averé que pour un paramètre de pénalité PEN=1.0, il n'y avait pas convergence vers la solution.

1.6.4 Variation du poids γ_i^p de chaque paire O/D et de chaque produit

Des tests ont été effectués sur le réseau "Dupuis" avec les caractéristiques suivantes :

- le nombre de produit est 2
- la capacité de chaque lien est 1.0

Les paramètres choisis pour les tests sur le réseau "Dupuis" sont les suivants:

Au niveau du problème global PL,

- Nombre maximum d'itérations : 20

PEN		1.0	1.25	1.5	1.75	2.0	pen=1.0 après 10 itérations pen=2.0
Nombre d'itérations effectués		20	20	20	4	5	13
Norme de la différence entre les multiplicateurs de deux itérations successives		0.39	0.47	0.76	0.038	0.011	0.022
SOLUTION g_i	g_{12}	59.98	59.11	60.07	60.75	60.83	60.83
	g_{13}	19.97	20.54	20.25	19.26	19.18	19.18
	g_{42}	19.97	21.07	20.25	19.26	19.18	19.18
	g_{43}	57.33	56.63	56.91	56.39	56.33	56.32

Tableau 1.2 : Variation du facteur de pénalité PEN

- Paramètre de précision ε_1 du critère d'arrêt sur les multiplicateurs : 0.05
- Paramètre de précision ε_2 du critère d'arrêt sur le flot prédit : 0.00005
- Paramètre de précision ε_3 du critère d'arrêt sur la demande prédite : 0.00005
- Paramètre de pénalité : 1.0 pour les dix premières itérations puis 2.0 pour les dix dernières itérations.

Au niveau du sous-problème PL(λ) où λ est fixé,

- Nombre maximum d'itérations : 20
- Paramètre de précision du critère d'arrêt : 0.00001
- Paramètre de précision pour la recherche unidimensionnelle : 0.001
- Valeur maximale de la demande prédite pour chaque paire O/D i de chaque produit p $g_i^{p\,max}$: 3.0

On fait varier le poids γ_i^p. Les résultats sont donnés au tableau (1.3). Là encore, on peut constater que la solution g_i^p n'est pas unique. On remarque que si les poids de chaque produit pour chaque paire O/D sont égaux, alors la solution est "symétrique", c'est-à-dire que la quantité maximale pouvant être envoyée pour une certaine paire O/D i est la même pour tous les produits de cette paire O/D. On peut également noter que les paires O/D i et les produits p ayant un poids plus élevé ont une solution g_i^p beaucoup plus élevée que les autres paires O/D des autres produits. Plus le poids est élevé par rapport aux autres produits et aux autres paires O/D, plus la quantité de produit g_i^p va être grande par rapport aux autres quantités de produit.

1.6.5 Variation de la demande maximale $g_i^{p\,max}$

ESSAI	PAIRE O/D	Nombre d'itérations	Norme de la différence entre les multiplicateurs de deux itérations successives	PRODUIT #1		PRODUIT #2	
				Poids du produit # 1	Solution g_i^1	Poids du produit # 2	Solution g_i^2
Essai # 1	1->2 1->3 4->2 4->3	2	0.0317	1.0 1.0 1.0 1.0	0.68 0.32 0.32 0.69	1.0 1.0 1.0 1.0	0.68 0.32 0.32 0.69
Essai # 2	1->2 1->3 4->2 4->3	20	0.5057	1.0 1.0 1.0 1.0	0.26 0.16 0.17 0.87	**3.0** **1.0** **1.0** **1.0**	**1.35** **0.16** **0.17** **0.87**
Essai # 3	1->2 1->3 4->2 4->3	20	0.22	**1.0** **3.0** **1.0** **1.0**	**0.39** **1.20** **0.50** **0.32**	1.0 1.0 1.0 1.0	0.39 0.12 0.50 0.32

Tableau 1.3 : Variation du poids des paires O/D i et des produits p

Des tests ont été effectués sur le réseau "Dupuis" avec les caractéristiques suivantes :

- le nombre de produit est 1
- la capacité de chaque lien est 1.0

Les paramètres choisis pour les tests sur le réseau "Dupuis" sont les suivants:

Au niveau du problème global PL,

- Nombre maximum d'itérations : 20
- Paramètre de précision ε_1 du critère d'arrêt sur les multiplicateurs : 0.05
- Paramètre de précision ε_2 du critère d'arrêt sur le flot prédit : 0.00005
- Paramètre de précision ε_3 du critère d'arrêt sur la demande prédite : 0.00005
- Paramètre de pénalité : 1.0 pour les dix premières itérations puis 2.0 pour les dix dernières itérations
- Poids de chaque paire O/D i pour chaque produit p : 1.0

Au niveau du sous-problème PL(λ) où λ est fixé,

- Nombre maximum d'itérations : 20
- Paramètre de précision du critère d'arrêt : 0.00001
- Paramètre de précision pour la recherche unidimensionnelle : 0.001

On fait varier la valeur maximale de la demande prédite pour chaque paire O/D i de chaque produit p : $g_i^{p\ max}$. Les résultats sont donnés au tableau (1.4). On remarque que plus la demande maximale est basse (soit près de la solution), plus le nombre d'itérations global requis est faible. Cela est dû au fait que la direction de descente du sous-problème PL(λ) avec λ fixé, prend une valeur entre 0 et $g_i^{p\ max}$ à chaque itération. De plus, la norme de la différence entre les multiplicateurs de deux itérations successives est plus faible pour une demande maximale près de la solution optimale. On doit fournir au programme une borne supérieure de la quantité de produit pour chaque paire O/D : on suppose donc cette information connue préalablement. En général, dans toute étude d'affectation de trafic, les

$g_i^{p\ max}$		20.0	10.0	5.0	3.0	2.0
Nombre d'itérations effectués		20	4	5	1	1
Norme de la différence entre les multiplicateurs de deux itérations successives		0.22	0.05	0.048	0.02	0.039
SOLUTION g_i	g_{12}	1.22	1.36	1.4	1.53	1.23
	g_{13}	0.65	0.63	0.61	0.48	0.77
	g_{42}	0.67	0.62	0.60	0.46	0.76
	g_{43}	1.34	1.37	1.39	1.52	1.25

Tableau 1.4 : Variation des demandes maximales $g_i^{p\ max}$

matrices de demandes pour chaque paire O/D et chaque produit sur la période considérée sont accessibles ou une prédiction des matrices de demande peut être calculée. Ces matrices nous donnent la valeur $g_i^{p\ max}$ pour chaque paire O/D i de chaque produit p. On peut alors déterminer la quantité de produit qui ne peut être affectée sur le réseau.
Notons par ailleurs que l'on peut forcer certaines paires O/D pour certains produits à ne pas dépasser une certaine quantité de produit en fixant $g_i^{p\ max}$ à la borne désirée. La solution obtenue par le calcul du flot maximal multiproduit g_i^p sera toujours inférieure à cette valeur $g_i^{p\ max}$.

1.6.6 Flot fixé sur chaque lien

Des tests ont été effectués sur le réseau "Dupuis" avec les caractéristiques suivantes :

- le nombre de produit est 1
- la capacité de chaque lien est 50.0

Les paramètres choisis pour les tests sur le réseau "Dupuis" sont les suivants:
Au niveau du problème global PL,

- Nombre maximum d'itérations : 20
- Paramètre de précision ε_1 du critère d'arrêt sur les multiplicateurs : 0.05
- Paramètre de précision ε_2 du critère d'arrêt sur le flot prédit : 0.00005
- Paramètre de précision ε_3 du critère d'arrêt sur la demande prédite : 0.00005
- Poids de chaque paire O/D i pour chaque produit p : 1.0

Au niveau du sous-problème PL(λ) où λ est fixé,

- Nombre maximum d'itérations : 20
- Paramètre de précision ε du critère d'arrêt : 0.00001
- Valeur maximale de la demande prédite pour chaque paire O/D i de chaque produit

$g_i^{p\ max}$: 20.0 puis 80.0 puis 100.0

Les résultats sont donnés au tableau (1.5). On remarque que les résultats obtenus avec les différentes valeurs maximales de la demande prédite et les différents flots fixés ne sont pas les mêmes.

1.6.7 Capacités dépendant du flot dans les deux directions

Des tests ont été effectués sur un réseau de 6 noeuds et 6 liens donné à la figure (1.3). Ce réseau a les caractéristiques suivantes :

- le nombre de produit est 1
- la capacité de chaque lien est 1.0
- les liens de 5 vers 6 et de 6 vers 5 ont une capacité conjointe de 1

Les paramètres choisis pour les tests sur le réseau "Dupuis" sont les suivants:

Au niveau du problème global PL,

- Nombre maximum d'itérations : 20
- Paramètre de précision ε_1 du critère d'arrêt sur les multiplicateurs : 0.05
- Paramètre de précision ε_2 du critère d'arrêt sur le flot prédit : 0.00005
- Paramètre de précision ε_3 du critère d'arrêt sur la demande prédite : 0.00005
- Poids de chaque paire O/D i pour chaque produit p : 1.0

Au niveau du sous-problème PL(λ) où λ est fixé,

- Nombre maximum d'itérations : 20
- Paramètre de précision du critère d'arrêt : 0.00001
- Paramètre de précision pour la recherche unidimensionnelle : 0.001

On fait varier le facteur de pénalité **pen** pour voir son influence sur la solution. Les résultats sont donnés au tableau (1.6). On remarque que la capacité conjointe limite la valeur optimale de la solution à la valeur 0.5.

$g_i^{p\ max}$		20	80		100
f_a pour tout arc a		0	20		0
Solution g_i	g_{12}	20	46.40	66.40	53.65
	g_{13}	20	13.68	33.68	28.66
	g_{42}	20	13.62	33.62	28.66
	g_{43}	20	46.04	66.04	71.69
Temps d'exécution			17.4		17.1

Tableau 1.5 : Flot fixé sur chaque lien

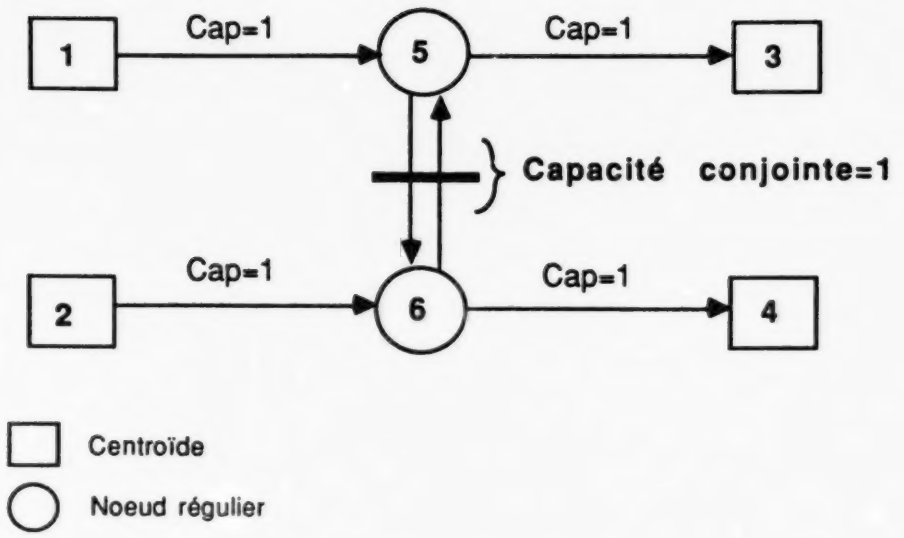


Figure 1.3 : Réseau test avec des capacités dépendant du flot dans les deux directions

PEN		1.0	2.0	pen=1.0 après 10 itérations pen=2.0
Nombre d'itérations effectués		20	1	20
Norme de la différence entre les multiplicateurs de deux itérations successives		1.953	1.00	1.00
SOLUTION	g_{14}	0.98	0.50	0.50
g_i	g_{23}	0.98	0.50	0.50

Tableau 1.6 : Capacités dépendant du flot dans les deux directions

1.6.8 Capacités différentes par combinaison de produit

Des tests ont été effectués sur un réseau avec 3 noeuds et 3 liens. En dupliquant les noeuds et les liens pour avoir une capacité par lien, on obtient le réseau donné à la figure (1.4). Les résultats sont similaires aux résultats obtenus dans un réseau usuel.

1.6.9 Borne inférieure sur la demande de certaines paires O/D pour certains produits : comparaison entre les deux méthodes de résolution

Des tests ont été effectués sur le réseau "Dupuis" avec les deux méthodes de résolution exposées à la section (1.5.4). Les caractéristiques du réseau "Dupuis" choisis pour les tests sont : - le nombre de produit est 3

- la capacité de chaque lien est 40.0

Les paramètres choisis pour les tests sur le réseau "Dupuis" sont les suivants :

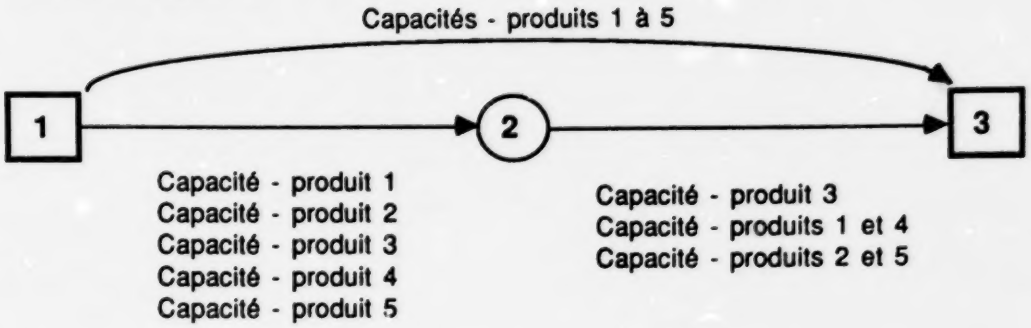
Au niveau du problème global PL,

- Nombre maximum d'itérations : 20
- Paramètre de précision ε_1 du critère d'arrêt sur les multiplicateurs : 0.05
- Paramètre de précision ε_2 du critère d'arrêt sur le flot prédit : 0.00005
- Paramètre de précision ε_3 du critère d'arrêt sur la demande prédite : 0.00005
- Paramètre de pénalité : 1.0 pour les dix premières itérations puis 2.0 pour les dix dernières itérations
- Poids de chaque paire O/D i pour chaque produit p : 1.0

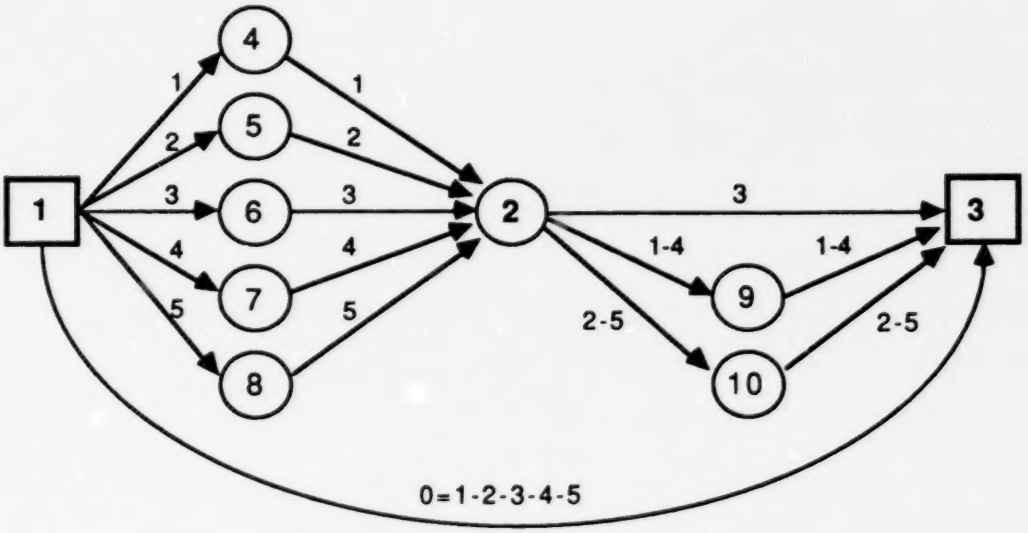
Au niveau du sous-problème PL(λ) où λ est fixé,

- Nombre maximum d'itérations : 20
- Paramètre de précision du critère d'arrêt : 0.00001
- Paramètre de précision pour la recherche unidimensionnelle : 0.001
- Valeur maximale de la demande prédite pour chaque paire O/D i de chaque produit p

Réseau original :



Réseau modifié :



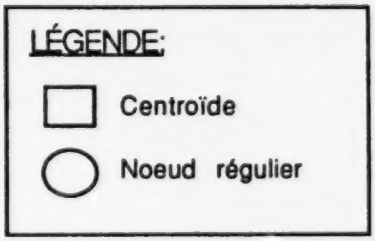


Figure 1.4 : Réseau test pour des capacités différentes par combinaison de produits

$g_i^{p\ max}$: $g_i^{1\ max} = 20.0$ pour toute paire O/D i

$g_i^{2\ max} = g_i^{3\ max} = 100.0$ pour toute paire O/D i

- Valeur minimale de la demande prédite pour chaque paire O/D i de chaque produit p

$g_i^{p\ min}$: $g_i^{1\ min} = 10.0$ pour toute paire O/D i

$g_i^{2\ min} = g_i^{3\ min} = 5.0$ pour toute paire O/D i

Les résultats sont donnés au tableau (1.7). On remarque que le nombre global d'itérations requis ainsi que la norme de la différence entre les multiplicateurs de deux itérations successives est moindre pour la deuxième méthode. De plus la solution obtenue par la première méthode viole les contraintes imposant une borne inférieure sur la demande prédite. En effet : $g_{12}^1 < g_{12}^{1\ min}$ $g_{13}^1 < g_{13}^{1\ min}$ $g_{42}^1 < g_{42}^{1\ min}$
Par contre la solution obtenue par la deuxième méthode ne viole aucune contrainte. Notons finalement que le temps d'exécution est deux fois moindre pour la deuxième méthode.

On peut donc en conclure que la deuxième méthode produit des meilleures solutions et qu'elle converge plus rapidement que la première méthode. Donc il est préférable d'utiliser la modification directe de l'algorithme de résolution du sous-problème $PL(\lambda_\ell)$ pour λ_ℓ fixé.

Les tests effectués sur les réseaux de petite taille montrent l'influence sur la solution optimale de la variation du facteur de pénalité PEN, de la variation du poids de chaque paire O/D i pour chaque produit p, et de la variation de la demande maximale $g_i^{p\ max}$.L'algorithme est complexe entre autres à cause des interactions entre les paramètres $g_i^{p\ max}$ et γ_i^p. On peut se demander si dans certains cas, une simplification de l'algorithme est possible. Si on a des bornes supérieures $g_i^{p\ max}$ qui limitent la demande prédite, alors il n'est pas possible de simplifier l'algorithme. Par contre, si les bornes choisies $g_i^{p\ max}$ sont suffisamment élevées pour ne pas limiter la demande prédite, on peut considérer que l'on a uniquement un seul produit, celui dont le poids est le plus important.

		Première méthode			Deuxième méthode		
Nombre d'itérations effectuées		20			13		
Norme de la différence entre les multiplicateurs de deux itérations successives		3 6			0 15		
Solution g_1		Produit 1	Produit 2	Produit 3	Produit 1	Produit 2	Produit 3
	g_{12}	9 26	20 49	20 49	11 01	14 55	14 55
	g_{13}	9 26	9 20	9 20	10 99	14 45	14 45
	g_{42}	9 26	9 24	9 24	11 0	14 45	14 45
	g_{43}	11 36	19 2	19 2	11 0	14 52	14 52
Temps d'exécution		39 69			21 67		

Tableau 1.7 : Borne inférieure sur la demande de certaines paires O/D pour certains produits : comparaison entre les deux methodes de resolution

Chapitre 2

MODELISATION PORTUAIRE

2.1 Motivation

Les ports constituent un aspect important de l'analyse du transport de marchandises à l'échelle nationale à cause de leur influence sur le reste du réseau. Que ce soit dans l'écoulement des marchandises de l'intérieur vers l'extérieur du pays, dans la distribution des marchandises de l'extérieur vers l'intérieur du pays ou bien dans le transport d'un point vers un autre point du pays par voie de navigation côtière (cabotage), les marchandises doivent emprunter les ports. Il est donc primordial de considérer le déroulement des activités s'effectuant à l'arrivée d'un port aussi bien qu'au départ d'un port, que ce soit pour l'importation ou pour l'exportation, ainsi que les transferts possibles de marchandises et l'entreposage dans le port. Ces différentes activités engendrent un certain nombre de coûts et de délais qui ne sont pas négligeables à l'échelle nationale. De plus on note des phénomènes de congestion dûs à la capacité restreinte de chaque port.

Pour que l'analyse globale soit consistante, il faut analyser les ports à un niveau de détail cohérant avec l'ensemble du réseau. Il s'agit donc de toujours tenir en compte que dans le réseau au complet, le but de l'analyse est stratégique.

Dans un premier temps, il faut trouver une représentation adéquate du réseau portuaire pour reproduire les activités essentielles influant de façon significative sur l'analyse globale du réseau. Cela implique définir précisément les caractéristiques du réseau portuaire.

Dans un deuxième temps, il faut déterminer quelle est l'analyse pouvant être effectuée dans un contexte stratégique national. Cette approche est totalement différente d'un approche opérationnelle à l'échelle urbaine.

La partie de l'analyse la plus délicate concerne l'entreposage des marchandises. Etant donné que l'on considère des périodes de temps bien déterminées et que l'on peut avoir entreposage entre deux périodes de temps consécutives, il est nécessaire de bien définir la méthode de traitement de ce qui sera entreposé.

2.2 Analyse et Modélisation d'un port

Objectifs d'une modélisation portuaire :

P. Wilness et E. Frankel (Wilness et al.) ont écrit différents articles sur l'analyse et la planification d'un port. Ils spécifient que les fonctions d'un port sont en général de transférer les produits entre les modes de transport maritimes et les modes de transport terrestres. Les critères ou les objectifs dans la modélisation d'un port et de ses opérations peuvent être :

1- Maximiser le flot à travers le port.
2- Maximiser le revenu des opérations du port.
3- Maximiser les profits provenant des opérations dans le port.
4- Remplir la capacité requise au coût minimal.
5- Obtenir un coût total de transport minimum en optimisant à la fois les composantes de système de transport et du port.

etc...

Ils indiquent également qu'une analyse portuaire s'effectue normalement dans une situation non-statique. On tient compte de la relation entre le temps nécessaire à l'expédition de la marchandise et les ressources disponibles pour obtenir un flot optimal. Les effets de coût et de délai sont donc des paramètres de grande importance dans la modélisation portuaire. Une analyse peut aussi être effectuée pour déterminer un optimum statique. Cet optimum définit la meilleure utilisation des ressources existantes. Dans le cas de notre modélisation, l'objectif 5 est celui qui nous intéresse.

Produits ou catégories de produits :

Dans le port, contrairement au reste du réseau, ce sont des produits qui circulent et non des véhicules. Les mouvements de la marchandise sont donnés en tonnes de produit et non en nombre de véhicules. De plus, tous les produits n'engendrent pas les mêmes coûts. On peut regrouper les produits avec des caractéristiques similaires de coût et de délai dans la même catégorie de produits. On peut définir les catégories générales de produit suivantes :

- marchandise générale
- grain
- charbon
- fer
- voiture ("rolling cargo")
- marchandise liquide
- "containers" (et les produits en sac)

etc....

Il est donc possible de définir différentes catégories de produits suivant les produits considérés dans l'analyse effectuée.

2.3 Mouvements de marchandises dans un port

Quels sont tous les mouvements possibles des marchandises depuis leur arrivée jusqu'à leur départ du port?
C'est une des premières questions à laquelle il faut apporter une réponse dans le processus de détermination de la représentation d'un port. Considérons les différentes étapes successives par lesquelles la marchandise doit passer à l'exportation, puis à l'importation.

Dans le cas de l'exportation, la marchandise arrive par véhicule, que ce soit par chemin de fer, par camion ou par cabotage. Il est possible qu'il y ait un certain temps d'attente dans le véhicule où la marchandise est arrivée. Après ce temps d'attente, la marchandise peut être déplacée dans son véhicule d'arrivée pour aboutir à l'endroit de déchargement. Les voies

d'accès du déplacement peuvent être soit uniquement pour les camions, soit uniquement pour les trains, soit permettre les deux modes. Puis vient un transfert possible à l'aide de certains équipements spécialisés pour acheminer la marchandise dans les entrepôts, si la marchandise ne peut être immédiatement envoyée sur le quai. Après ce délai possible d'entreposage (pour attendre le prochain bateau ou que le matériel ou les ressources humaines soient disponibles) la marchandise est acheminée vers le quai d'embarquement pour son chargement dans le bateau. Une fois embarquée, la marchandise est envoyée vers son lieu de destination, à l'extérieur du pays ou vers un autre port du pays.

Dans le cas de l'importation, la marchandise arrive au port par bateau de son lieu d'origine qui est un autre point soit du pays, soit à l'extérieur du pays. La marchandise est déchargée du bateau. Vient ensuite un entreposage possible de la marchandise si les ressources humaines ou si le matériel (le camion ou le train dans lequel la marchandise va être véhiculée à travers le réseau ou les équipements à l'intérieur du port) ne sont pas disponibles pour faire sortir la marchandise du port. Après ce délai possible d'entreposage, la marchandise est acheminée vers le lieu de chargement puis est placée dans le véhicule destiné à son transport à l'intérieur du pays. Il est également possible que la marchandise attende dans son véhicule avant de sortir du port. La marchandise sort par chemin de fer, par camion, ou par cabotage.

Dans cette analyse de mouvements dans un port, il a été fait abstraction de la notion de terminal ainsi que des mouvements de marchandise d'un terminal vers un autre terminal. En réalité, il peut y avoir mouvement de la marchandise d'un terminal à l'autre dans des équipements particuliers du port si un terminal a atteint sa capacité . Ces mouvements sont possibles à plusieurs moments : après le déchargement de la marchandise, à l'entreposage, etc....

La figure 2.1 montre le cheminement de la marchandise en distance et en temps dans un certain terminal du port. Notons que dans cette figure, les liens 1 à 11 représentent les liens d'exportation tandis que les liens 12 à 22 représentent les liens d'importation. Pour les liens 1 à 4, 17 à 21, et 10, la marchandise est dans le véhicule de transport. Pour les liens 8,9,12,13, la marchandise est dans le bateau. Pour les liens 6 et 15, la marchandise est dans le lieu d'entreposage. Notons également que pour les liens 4 et 17 correspondant au déchargement (chargement) du véhicule, on peut avoir une combinaison du mode routier et du mode ferroviaire.

2.4 Simplifications apportées au modèle

2.4.1 Première simplification

A partir de la connaissance des mouvements de la marchandise dans le port, on peut définir une représentation générale d'un port. Le réseau portuaire doit inclure l'exportation et l'importation et doit tenir en compte de l'aspect statique du modèle.

Il s'agit donc de simplifier la représentation portuaire précédente pour intégrer le réseau portuaire dans le réseau multimode et multiproduit global. Pour des raisons de compatibilité, on cherche à représenter le port de la même façon que le reste du réseau, c'est-à-dire à avoir les mêmes caractéristiques. Le reste du réseau est composé de noeuds,de centroïdes (noeuds où origine ou aboutit la demande), de liens avec une capacité, une fonction de coût associée à chaque lien et une fonction de pénalité associée à chaque noeud. On garde la notion de mode de transport à l'entrée du port ainsi que la notion de produit. De plus, puisque l'analyse au niveau du réseau global est faite de façon statique, il faut supprimer la notion de temps de la représentation précédente.

La figure 2.2 montre la représentation générale d'un port (importation et exportation). Notons que l'on peut accéder au port par un des trois modes de transport : routier, ferroviaire, ou cabotage. Le noeud 14 est un noeud fictif modélisant l'accès du reste de réseau. Si on intègre le réseau portuaire au réseau global, les liens routiers se connectent directement au

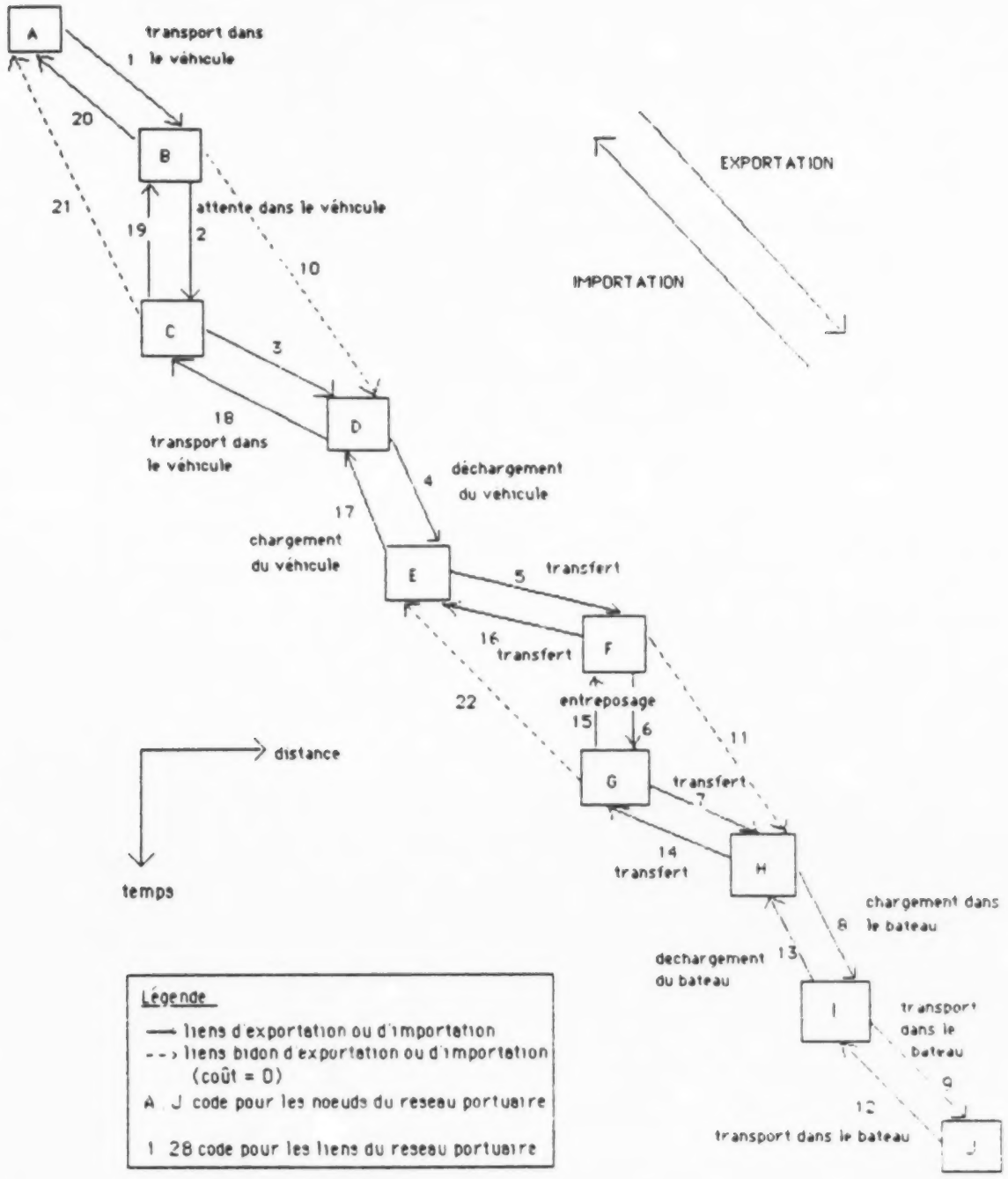


Figure 2.1 : **Mouvements de la marchandise dans un port**

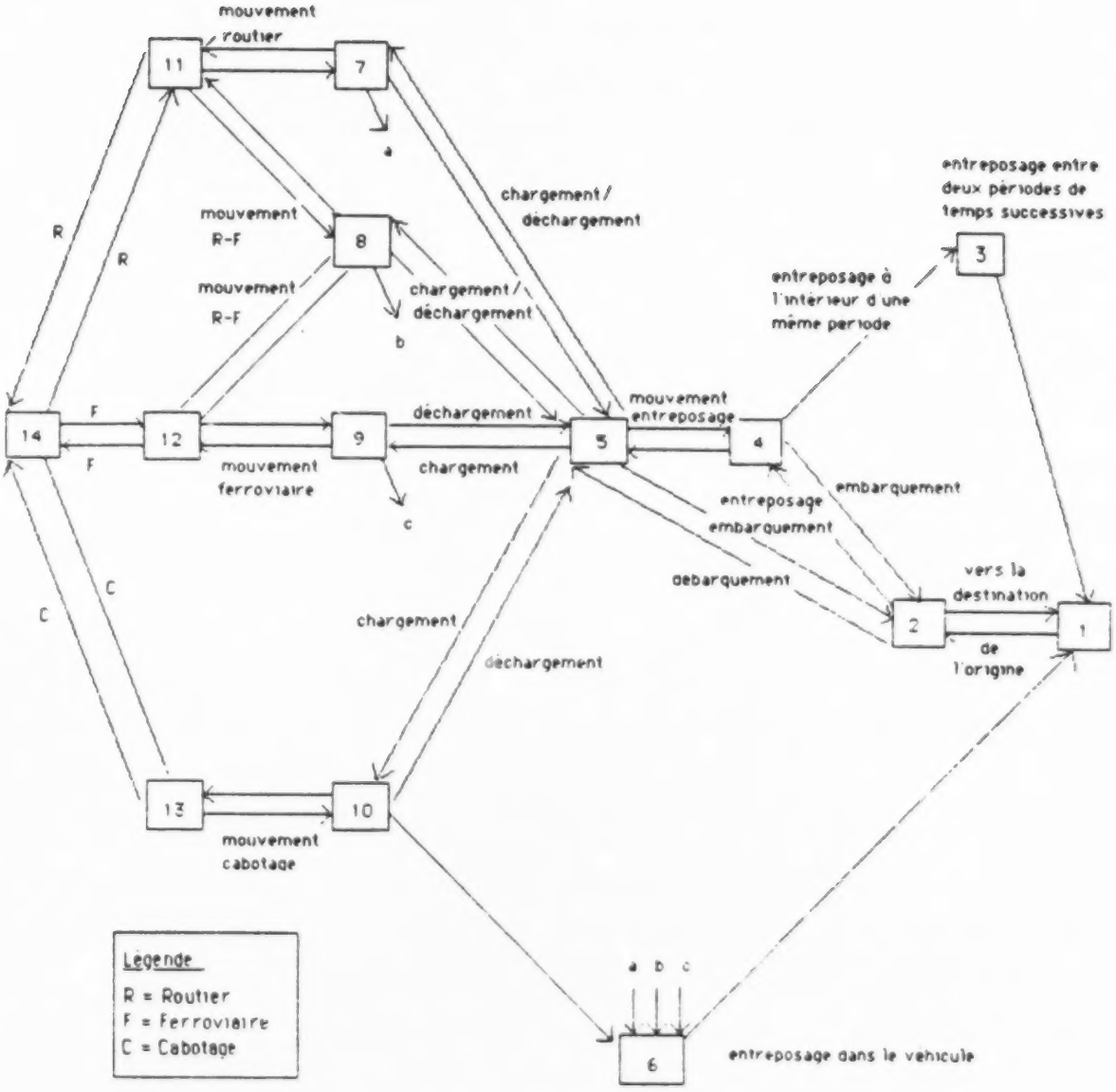


Figure 2.2 Représentation générale d'un port importation et exportation

noeud 11, les liens ferroviaires au noeud 12, et les liens de cabotage au noeud 13. Puis on a certains liens représentant les mouvements de la marchandise pour chacun des modes. Le noeud 8 permet de modéliser la combinaison permise du mode ferroviaire et routier. On a des liens de déchargement (chargement) de la marchandise et des liens d'entreposage dans le véhicule si les ressources humaines ou si le matériel ne sont pas disponibles. Tous les liens de déchargement (chargement) aboutissent (proviennent) du noeud 5. A partir du noeud 5, on a un lien d'embarquement (débarquement) et un lien de mouvement vers le (du) lieu d'entreposage. Du lieu d'entreposage représenté par le noeud 4, on peut avoir embarquement (débarquement) de la marchandise ou entreposage si durant la période d'analyse considérée, il n'a pas été possible d'envoyer toute la marchandise. Le noeud 3 permet de modéliser l'entreposage entre deux périodes d'analyse consécutives pour le cas où la marchandise ne peut pas être envoyée dans une période d'analyse.

2.4.2 Deuxième simplification

2.4.2.1 Réprésentation du réseau portuaire

On peut simplifier encore la représentation de la figure 2.2 car le réseau portuaire est encore trop détaillé par rapport à l'ensemble du réseau de transport sur tout le pays. On peut en effet agréger tous les liens représentant les mouvements de la marchandise dans le véhicule ainsi que le déchargement (chargement) de la marchandise. On modèlise par un seul lien pour chaque mode de transport l'arrivée (départ) de la marchandise au port ainsi que le déchargement (chargement) du véhicule. L'entreposage de la marchandise dans le véhicule entre deux périodes d'analyse consécutives est également négligeable par rapport à l'entreposage dans les entrepôts entre ces deux mêmes périodes d'analyse consécutives.

Le réseau portuaire se simplifie donc par 3 noeuds et 12 liens : 6 liens relient les 3 noeuds du port et 6 autres liens servent d'accès au port (deux liens d'accès par mode de transport, un lien en direction du port et un autre en provenance du port). On accède (sort) donc au (du) port par un des 6 liens d'accès dépendant du mode de transport utilisé (routier

ferroviaire, ou cabotage). Puis on peut embarquer (débarquer) directement la marchandise ou embarquer (débarquer) "indirectement" la marchandise : dans ce deuxième cas, l'embarquement (débarquement) de la marchandise est précédé (suivi) par un entreposage.

La figure 2.3 montre le processus de simplification du réseau portuaire, c'est-à-dire quelles ont été les simplifications apportées pour passer de la représentation de la figure 2.2 à un réseau portuaire comportant uniquement 3 noeuds et 12 liens. La figure présente trois réseaux correspondant aux étapes de simplification.

La figure 2.4 montre la représentation simplifiée d'un port à l'exportation, à l'importation, puis à la fois à l'exportation et à l'importation. Dans cette figure, les noeuds 1,2,3,4 sont les noeuds du port et les noeuds A,B,C sont des centroïdes, c'est-à-dire des noeuds où origine ou aboutit la demande de certains produits. Le centroïde A est l'origine et/ou la destination du port. Le centroïde B est l'origine et/ou la destination à l'extérieur du port. Le centroïde C est l'origine du port pour la marchandise qui n'a pas pu être envoyée aux périodes d'analyse précédentes. Les noeuds 1,2 et 3 sont les trois noeuds réguliers du port découlant de la simplification du réseau. On a ajouté un noeud 4 pour pouvoir modéliser correctement les mouvements correspondant à la réexpédition de la marchandise à partir du centroïde origine C à la prochaine période d'analyse. Nous expliciterons à la section sur l'entreposage la raison de l'ajout de ce noeud 4. Notons aussi la présence de liens de cabotage originant ou aboutissant au noeud 2 : ces liens modélisent les mouvements de marchandise d'un port à un autre port du pays par mode cabotage.

2.4.2.2 Mouvements de la marchandise

A partir de cette représentation, retraçons certains mouvements possibles de la marchandise à l'exportation, puis à l'importation.

A l'exportation, si la destination est le port lui-même, c'est-à-dire si la demande aboutit au centroïde A, alors on a les mouvements suivants :

- le mouvement noeud 1→noeud 2→noeud A représente un embarquement direct de la

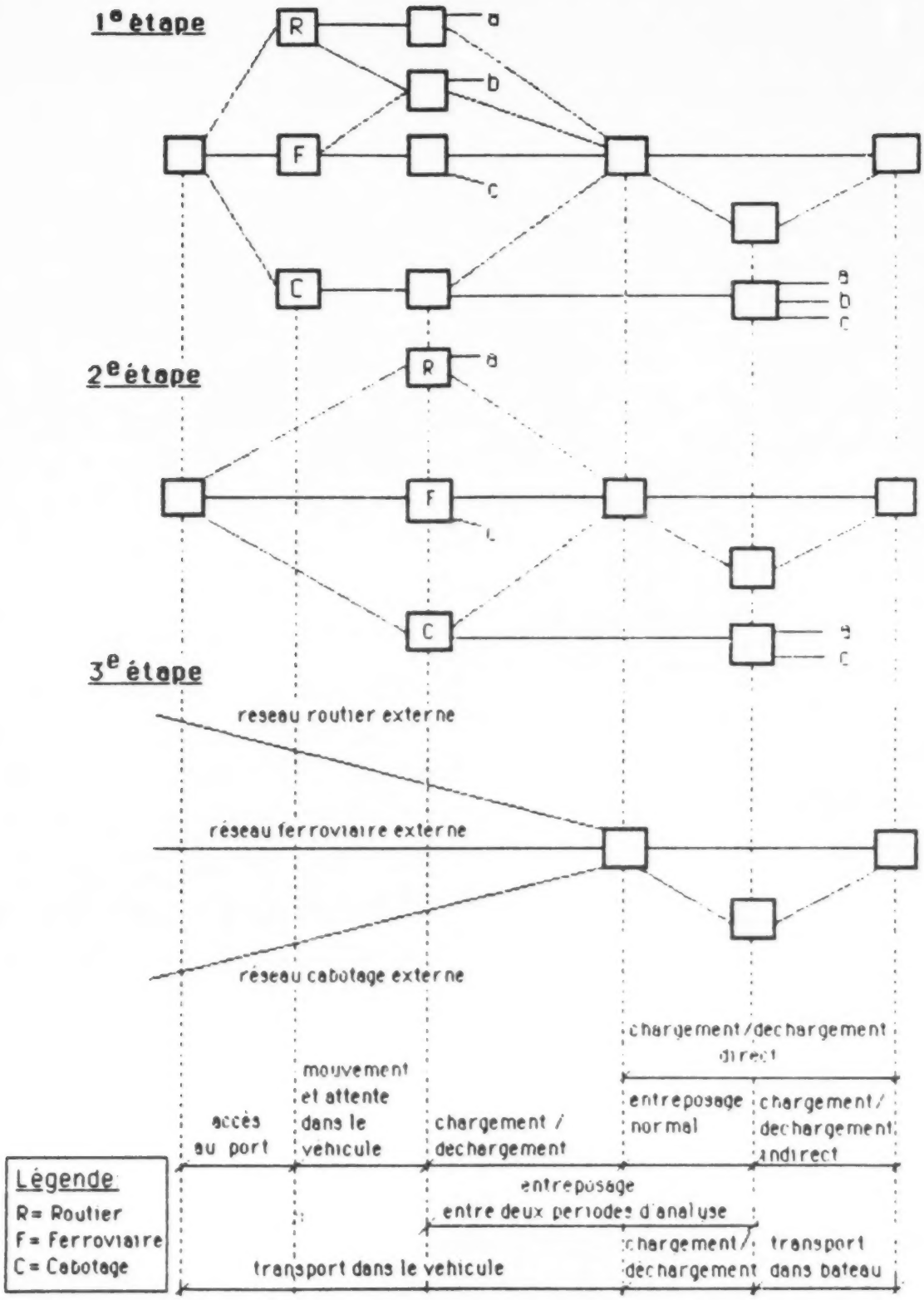


Figure 2.3 : Etapes de simplification du réseau portuaire

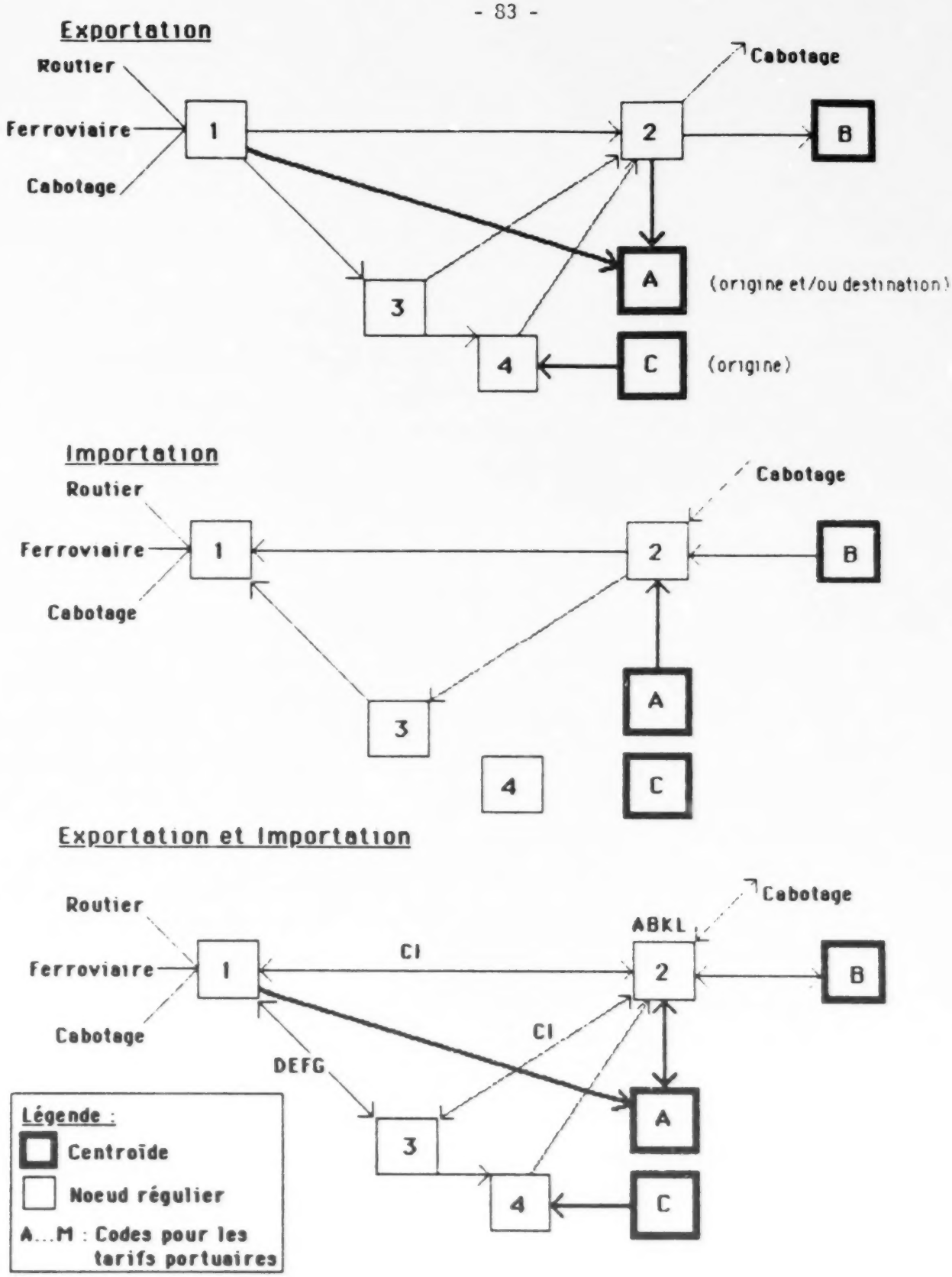


Figure 2.4 : Représentation simplifiée d'un port

marchandise sans passer par un entreposage.

- le mouvement noeud 1→noeud 3→noeud 2→noeud A représente un embarquement indirect de la marchandise, c'est-à-dire en passant par un entreposage.
- le mouvement noeud 1→noeud A représente un entreposage entre deux périodes d'analyse dans le cas où les ressources humaines ou matérielles ne sont pas disponibles d'ici la fin de la période d'analyse.
- le mouvement noeud C→noeud 4→noeud 2→noeud A représente un mouvement pour la marchandise qui n'avait pas pu être envoyée à la période d'analyse précédente.

A l'exportation, si la destination est un point à l'extérieur du port (dans un autre pays), c'est-à-dire si la demande aboutit au centroïde B, alors on a les mouvements suivants :

- le mouvement noeud 1→noeud 2→noeud B représente un embarquement direct de la marchandise sans passer par un entreposage.
- le mouvement noeud 1→noeud 3→noeud 2→noeud B représente un embarquement indirect de la marchandise, c'est-à-dire en passant par un entreposage.
- le mouvement noeud C→noeud 4→noeud 2→noeud B représente un mouvement pour la marchandise qui n'avait pas pu être envoyée à la période d'analyse précédente.

A l'exportation, si la destination est un autre port du pays, c'est-à-dire si la demande utilise les liens de cabotage reliant chaque port, alors on a les mouvements suivants :

- le mouvement noeud 1→noeud 2→cabotage représente un embarquement direct de la marchandise sans passer par un entreposage.
- le mouvement noeud 1→noeud 3→noeud 2→cabotage représente un embarquement indirect de la marchandise, c'est-à-dire en passant par un entreposage.
- le mouvement noeud C→noeud 4→noeud 2→cabotage représente un mouvement pour la marchandise qui n'avait pu être envoyée à la période d'analyse précédente.

A l'importation, si l'origine est le port lui-même, c'est-à-dire si la demande origine du centroïde A, alors on a les mouvements suivants :

- le mouvement noeud A→noeud 2→noeud 1 représente un débarquement direct de la marchandise sans passer par un entreposage.
- le mouvement noeud A→noeud 2→noeud 3→noeud 1 représente un débarquement indirect de la marchandise c'est-à-dire en passant par un entreposage.

A l'importation, si l'origine est un point à l'extérieur du port (dans un autre pays), c'est-à-dire si la demande origine du centroïde B, alors on a les mouvements suivants :

- le mouvement noeud B→noeud 2→noeud 1 représente un débarquement direct de la marchandise sans passer par un entreposage.
- le mouvement noeud B→noeud 2→noeud 3→noeud 1 représente un débarquement indirect de la marchandise, c'est-à-dire en passant par un entreposage.

A l'importation, si l'origine est un autre port du pays, c'est-à-dire si la demande utilise les liens de cabotage reliant chaque port, alors on a les mouvements suivants :

- le mouvement cabotage→noeud 2→noeud 1 représente un débarquement direct de la marchandise sans passer par un entreposage.
- le mouvement cabotage→noeud 2→noeud 3→noeud 1 représente un débarquement indirect de la marchandise, c'est-à-dire en passant par un entreposage.

Tous les mouvements possibles n'ont pas été répertoriés car le modèle offre la possibilité d'empêcher certains mouvements pour certaines catégories de produits à l'aide de fonctions de pénalité ou de fonctions de délais comme nous le verons aux prochaines sections.

2.4.2.3 Inter-relation entre les différents ports

Considérons maintenant plusieurs ports. Comment sont-ils connectés ? Tous les ports doivent être connectés pour pouvoir accéder aux autres ports par voie de navigation côtière. Pour modéliser de façon adéquate le cabotage, on ajoute un noeud intermédiaire à chaque port ainsi qu'un lien reliant le noeud 2 à ce nouveau noeud intermédiaire. On relie par la suite tous les noeuds intermédiaires de chaque port avec ceux des deux ports adjacents.

La figure 2.5 montre la connection de trois ports par l'ajout des noeuds 5 pour chacun des ports et des liens reliant les noeuds 2 de chaque port avec ces noeuds 5 ajoutés. La figure montre également les liens reliant les noeuds 5 entre eux. Notons que le centroïde B est une destination à l'extérieur du pays et que la marchandise peut utiliser un des trois ports pour y acheminer la marchandise.

Bien qu'il est probable que les ports ne sont pas tous reliés entre eux dans la réalité, on connecte dans le modèle tous les ports pour fin de généralité.
Notons également que la représentation portuaire peut varier légèrement pour des raisons géographiques. La figure 2.6 montre la représentation de deux ports reliés par voie de navigation interne. Dans cette figure, le port 1 est au fond d'une lagune et peut accéder au port 2 situé sur la rivière ou aller directement vers une autre destination modélisée par le centroïde B .

2.4.2.4 Fonctions de transfert :

On remarque que les mouvements possibles sont nombreux et que le nombre de noeuds et de liens dans le réseau portuaire est restreint. Pour modéliser d'une façon adéquate tous les mouvements permis et non permis, il est donc nécessaire d'avoir des fonctions qui déterminent les coûts engendrés par ces mouvements ou qui empêchent certains mouvements. Cela permettra de simuler les flots de marchandises à travers tout le réseau, de détermine

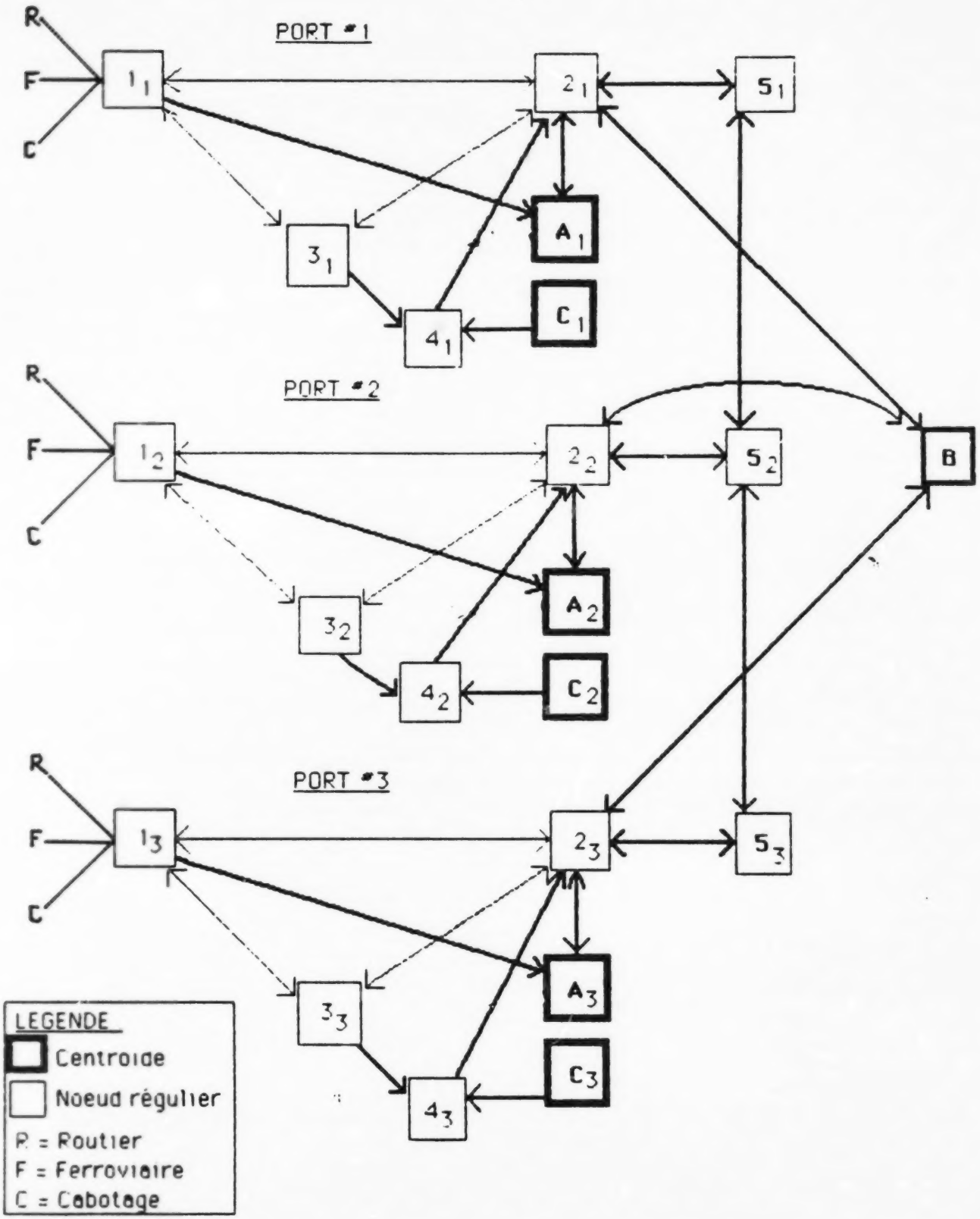


Figure 2.5 : Représentation de la connection de trois ports

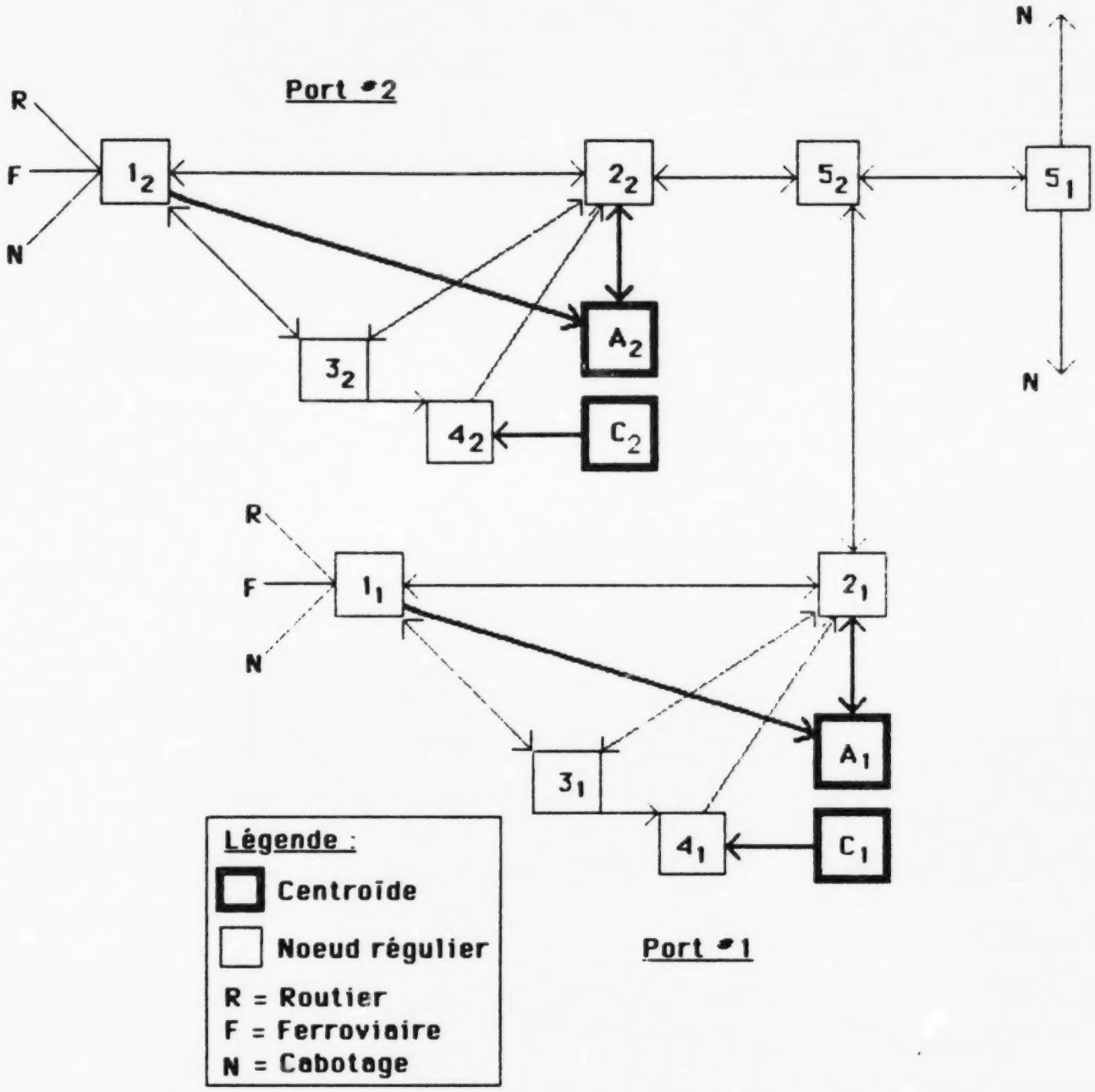


Figure 2.6 : **Représentation de la connection de deux ports sur navigation interne**

par les coûts d'embarquement quelle quantité de marchandise sera effectivement envoyée durant la période d'analyse et quelle quantité sera entreposée.

Pour ce faire, on utilise donc des fonctions de transferts aux différents noeuds du réseau portuaire pour les différents produits ou catégories de produit considérés. Une fonction de transfert est une fonction de pénalité associée à un noeud donné pour passer d'un lien à un autre lien par ce noeud. Une fonction de transfert est basée sur le volume des marchandises et peut être linéaire ou convexe. On a deux types de fonctions de transfert: la première est une fonction avec un coût fini qui modélise un coût ou un délai de transfert. La deuxième est une fonction avec un coût infini qui modélise un délai infini de transfert (c'est-à-dire une impossibilité de passer par ce noeud). Certaines fonctions de transfert dans le réseau portuaire pour l'exportation et pour l'importation sont les suivantes :

Fonction de transfert au noeud 1 :

1) lien routier-noeud 1-lien (1-2) lien routier-noeud 1-lien (1-3)
lien ferroviaire-noeud 1-lien (1-2) lien ferroviaire-noeud 1-lien (1-3)
lien cabotage-noeud 1-lien (1-2) lien cabotage-noeud 1-lien (1-3)

Ces fonctions de transfert correspondent à un délai fini pour les catégories de produits permises.
Ces six pénalités de transfert à l'exportation modélisent à la fois le mouvement de la marchandise, l'attente dans le véhicule de transport et le déchargement du véhicule.

2) lien (2-1)-noeud 1-lien routier lien (3-1)-noeud 1-lien routier
lien (2-1)-noeud 1-lien ferroviaire lien (3-1)-noeud 1-lien ferroviaire
lien (2-1)-noeud 1-lien cabotage lien (3-1)-noeud 1-lien cabotage

Ces fonctions de transfert correspondent à un délai fini pour les catégories de produits permises.
Ces six pénalités de transfert à l'importation modélisent à la fois le déchargement de la marchandise, l'attente dans le véhicule de transport et le mouvement dans le véhicule.

Fonction de transfert au noeud 2 et au noeud 3 :

lien (1-2)-noeud 2-lien (2,A)
lien (1-2)-noeud 2-lien (2,B)
lien (1-3)-noeud 3-lien (3,A)
lien (1-3)-noeud 3-lien (3,B)
lien (3-2)-noeud 2-lien (2,A)
lien (3-2)-noeud 2-lien (2,B)

Pour chacun de ces transferts,on peut avoir une fonction de pénalité pour tous les produits ou catégories de produit.

Fonction de transfert au noeud 4 :

lien (3-4)-noeud 4-lien (4,2)

Ces fonctions de pénalité sont des fonctions avec un délai infini de transfert pour toutes les catégories de produits car le modéle ne permet pas ce mouvement.

2.4.2.5 Fonctions de délai basées sur le volume :

Chaque lien dans le réseau portuaire a un coût (fonction de délai) et une capacité. Les fonctions sont soit convexes, soit linéaires. La capacité est définie comme le nombre maximum de tonnes de marchandise qui peut traverser le lien durant la période considérée. Les capacités sont modélisées comme des fonctions linéaires par morceaux. Pour séparer la quantité de marchandises à envoyer sur un arc de la quantité à envoyer sur l'arc parallèle (si la quantité de marchandise excède la capacité), nous modélisons la fonction du premier arc par une fonction convexe. La fonction de l'arc parallèle est une fonction linéaire à partir du point correspondant à la capacité du premier arc. Etant donné que le modèle envoie la marchandise sur les arcs de coûts minimuns et que à partir du point de capacité, la deuxième fonction est inférieure à la première, alors la quantité de marchandise en excès sera envoyée sur l'arc de décharge. La figure (2.7) illustre ce principe.

Dans le cas du transport de marchandise au Brésil, on modélise les tarifs associés aux différentes manipulations par des coûts de délai et non par des coûts de perte d'énergie. Les raisons sont d'une part que la perte diffère suivant que l'énergie utilisée est électrique ou

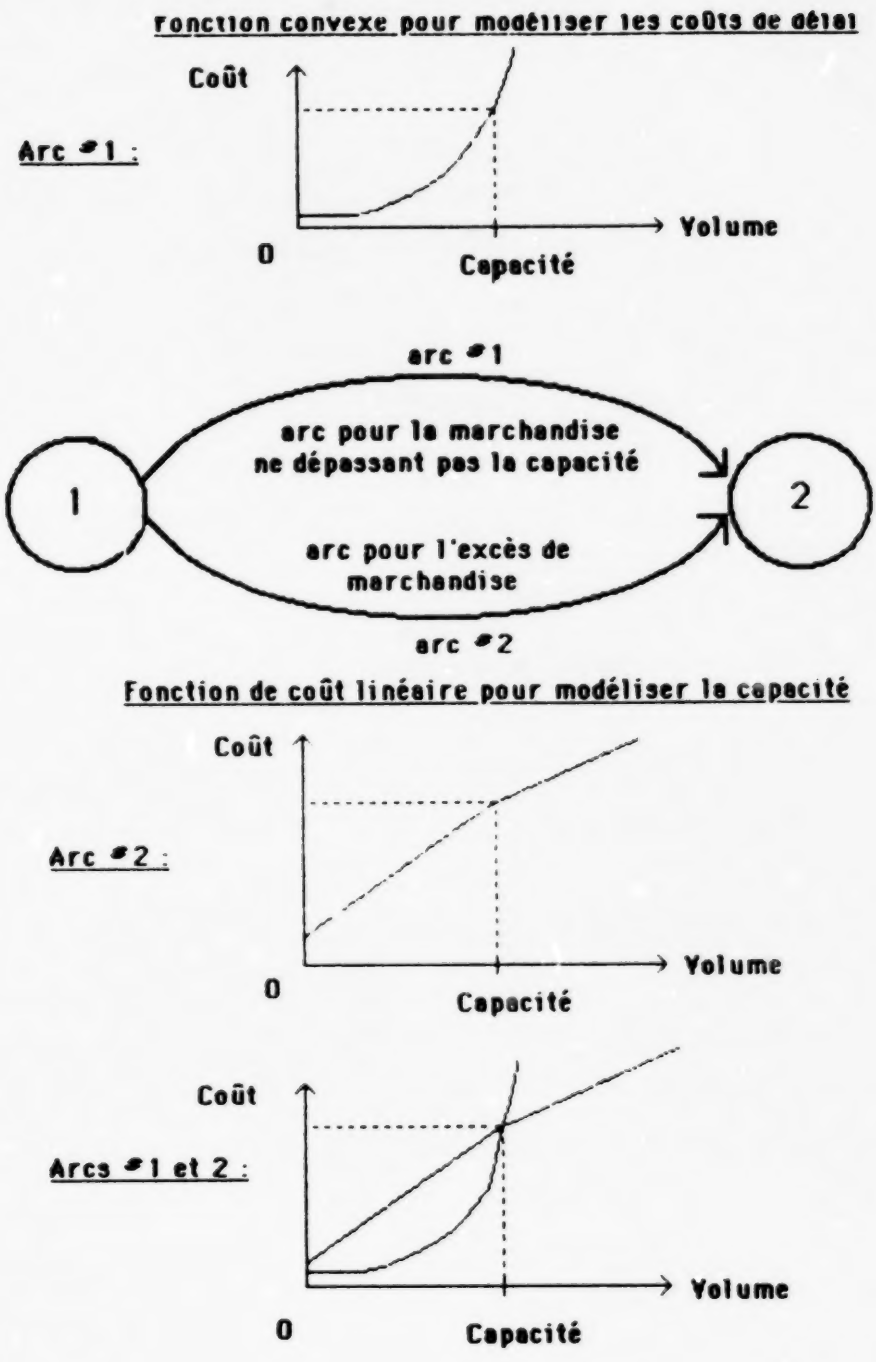


Figure 2.7 : Fonction de coût pour modéliser la capacité

diesel, d'autre part que les coûts engendrés par les pertes d'énergie sont plus faibles que ceux dûs aux délais et finalement que la perte d'énergie dans les ports est beaucoup moindre que dans le reste du réseau.

Les délais considérés sont de trois ordres :

1) **par bateau** : pour le chargement et le déchargement de la marchandise par catégorie de produits.

2) **par qual** : pour le chargement et le déchargement de la marchandise (temps moyen de chargement divisé par le tonnage moyen). La fonction de délai inclut le temps d'attente sur le quai.

3) **par marchandise enveloppée ("wrapping cargo")** : cela donne la productivité moyenne.

Dans le cas du réseau brésilien, les tarifs portuaires sont sous la forme de tables. Les catégories de tables sont les suivantes :

Table A- Utilisation du port

Table B- Utilisation des quais pour accoster le bateau.

Table C- Travaux du contremaître

Table D- Entreposage interne

Table E- Entreposage externe

Table F- Entreposage dans les entrepôts généraux.

Table G- Entreposage spécial (refrigérateur,silos,matériel corrosif,inflammable...)

Table H- Transport (chargement, déchargement et transport par chemin de fer)

Table I- Arrimage des embarcations (transfert de marchandises pour le cabotage)

Table J- Suppression de l'apparaillage portuaire.

Table K- Mouvement des embarcations dans les eaux portuaires

Table L- Suppression de l'eau dans les embarcations

Table M- Services accessoires

Table N- Mouvements de la marchandise des quais d'embarquement.

Dans le cas du transport de marchandises au Brésil, les noeuds (fonction de transfert) ou liens (fonction de délai) où ses différents tarifs s'appliquent sont indiqués à la figure (2.4) dans le schéma de l'exportation et de l'importation.

Dans le réseau portuaire, nous pouvons donc associer un coût à tout lien du réseau portuaire pour tenir compte de la capacité restreinte des ports. Ces fonctions de délai peuvent différer d'une catégorie à une autre de produits.

2.5 Entreposages

Une des caractéristiques du réseau portuaire est la présence de points d'entreposage : en effet, la marchandise peut être entreposée soit à l'intérieur du véhicule de transport, soit dans des entrepôts spécialisés pour le type de produit considéré.

Le problème provient du fait que l'étude de la planification du réseau s'effectue de façon statique alors que la notion d'entreposage est en elle-même un concept dynamique. L'analyse du trafic des marchandises s'effectuant donc sur une période de temps déterminée préalablement, il est nécessaire de faire une analyse multipériode dans le but de déterminer quelle quantité de marchandise n'a pas pu être envoyée durant la période considérée pour pouvoir envoyer cet excès de marchandises à la prochaine période d'analyse.

2.5.1 Entreposage dans le véhicule de transport

L'entreposage dans les véhicules de transport n'a pas été modélisé dans un but de simplication du modèle. Cet entreposage est à un niveau trop détaillé pour une analyse stratégique sur tout le réseau.

2.5.2 Entreposage dans le port à l'intérieur d'une même période d'analyse

A l'exportation, les liens (1,2) et (1,3) correspondant aux activités de chargement de la marchandise partagent une même capacité. On associe une fonction de délai à ces deux liens pour toutes les catégories de produit afin de déterminer la quantité de marchandise qui va être envoyée sur le lien (1,2) et la quantité qui va être envoyée sur le lien (1,3). De même à l'importation, les liens (2,1) et (2,3) correspondant aux activités de déchargement de la marchandise partagent une même capacité. On associe une fonction de délai à ces deux liens pour toutes les catégories de produit afin de déterminer la quantité de marchandise qui va être envoyée sur le lien (2,1) et la quantité qui va être envoyée sur le lien (2,3). Il faut calibrer adéquatement les fonctions pour modéliser le partage des marchandises entre le chargement (déchargement) et l'entreposage. La calibration se fait suivant le principe illustré à la figure (2.7).

2.5.3 Entreposage dans le port entre deux périodes d'analyse consécutives

Etant donné que le modèle global sur l'ensemble du réseau est un modèle statique basé sur l'étude d'une période donnée, il est nécessaire de prévoir une analyse multipériode. Nous devons tenir en compte l'emmagasinage des différents produits entre deux périodes de temps consécutives. La difficulté provient de ce que les règles que l'usager veut se définir peuvent varier d'une étude à l'autre. Donc il est préférable de déterminer des procédures générales plutôt que d'inclure des règles spécifiques qui pourraient ne pas s'appliquer dans certains cas.

La méthode de traitement de l'entreposage est différente à l'exportation et à l'importation. **A l'exportation,** la solution adoptée pour traiter l'entreposage de la marchandise est la suivante : Le lien (1,A) est le lien de décharge pour envoyer la quantité de marchandise excédant la somme des capacités des liens (1,2) et (1,3). En d'autres mots, cela correspond au cas où la capacité du port ne permet pas d'envoyer directement ou "indirectement" toute la marchandise à la destination (la destination pouvant être le port ou un point à l'extérieur du pays). L'excès ("overflow") utilise alors le lien (1,A). La quantité de marchandise qui n'a pas pu être envoyée aboutit au centroïde du port : le centroïde A. Il faut

calibrer correctement les fonctions de délai pour déterminer la quantité de marchandise envoyée sur les liens (1,2) et (1,3) de la quantité envoyée sur le lien (1,A). Le principe est celui de la figure (2.7) et peut être généralisé au cas de trois arcs. Nous utilisons le centroïde C (voir figure (2.4)) pour envoyer l'excès de marchandise à la prochaine période d'analyse. Pour la prochaine période d'analyse, il est donc nécessaire de modifier les matrices Origine/Destination (O/D) pour ajouter le centroïde C à la liste des centroïdes origines avec les destinations appropriées pour la marchandise qui n'a pu être envoyée à la période d'analyse précédente. On a donc une demande originant du centroïde C pour les catégories de produit qui n'ont pu être envoyées à la période d'analyse précédente. La marchandise repart du centroïde C puis passe par le noeud 4 pour atteindre sa destination finale. La raison pour laquelle on ne connecte pas directement le centroïde C au noeud 3 est pour pouvoir traiter durant la même affectation de trafic les quantités de marchandise en excédant de la période d'analyse précédente et les quantités de marchandise originant de l'intérieur du pays pour la période d'analyse courante.

La quantité de tonnes de produit pouvant être envoyée et la quantité devant être entreposée entre deux périodes d'analyse peut être déterminée au préalable (par un pré-traitement).

Nous pouvons déterminer auparavant la quantité de produit à envoyer au port et la quantité qui doit être entreposée entre deux périodes d'analyse. Un calcul du flot maximal de produit pouvant être envoyé sur le réseau par la méthode explicitée au Chapitre 1 nous donne l'information désirée. Nous pouvons alors dissocier dans les matrices O/D la quantité de marchandise à entreposer à la fin de la période d'analyse de la quantité à envoyer à sa destination : nous pouvons alors modifier les matrices O/D pour que la marchandise termine à un certain port (au centroïde A de ce port). On peut aussi laisser le modèle choisir la quantité et le port par lequel la marchandise peut passer. Pour certains produits, nous pouvons souhaiter avoir la première situation, pour d'autres produits la deuxième situation.

A l'importation, le problème est plus délicat car il faudrait un lien pour la marchandise qui ne peut être envoyée durant la période d'analyse du centroïde A (port) ou B (point à l'extérieur du pays) à toutes les destinations possibles. Or, si le nombre de centroïdes dans le réseau est élevé, cela implique un trop grand nombre de liens à ajouter.

Dans le cas de **l'importation**, nous devons effectuer un pré-traitement ("pre-processing") pour déterminer la quantité de marchandise pouvant être envoyée sur le réseau de transport Le pré-traitement s'effectue en deux étapes :

étape 1 : pour chaque port, nous vérifions si la quantité de marchandise originant du port peut sortir du port étant donné la capacité de ce port.

étape 2 : pour chaque centroïde de type B, nous vérifions si la quantité de marchandise originant de l'extérieur peut passer par un des ports. Nous calculons donc la capacité totale de tous les ports et nous vérifions si la quantité de marchandise totale n'excède pas cette capacité. Puis on effectue un calcul de flot maximal multiproduit (méthode du Chapitre 1) pour vérifier si la marchandise peut effectivement s'écouler à travers le réseau. Il se pourrait en effet que les capacités sur les liens empêchent la marchandise d'aller d'une origine à une destination même si la capacité totale de tous les ports peut contenir toutes les marchandises.

Dans le cas du réseau brésilien, nous considérons des périodes d'analyse d'au moins trois mois. Lorsque nous devons étudier la prochaine période, nous devons tenir compte des tonnes de produits entreposées à la fin de la période précédente. Dans une analyse multipériode, par exemple dans l'analyse des mouvements des grains sur une année, l'usager doit effectuer les pré-traitements adéquats pour l'analyse qu'il désire effectuer et modifier les matrices O/D à chaque période pour considérer les grains entreposés après chaque trois mois.

2.6 Conclusion

A partir de l'étude des mouvements de marchandise, on a déterminé une représentation statique du réseau portuaire que l'on a simplifié pour avoir un niveau de détail cohérent dans le cadre d'une étude stratégique à l'échelle d'un pays.

La première conclusion que l'on peut tirer est que le port n'utilise aucune structure particulière : on utilise les mêmes caractéristiques que pour le reste du réseau, soit des noeuds, des liens et des centroïdes. Nous avons des fonctions de transfert possiblement à chaque noeud du réseau portuaire pour chaque paire de liens et pour chaque catégorie de produit dans le but de modéliser les coûts et les délais engendrés par les diverses manipulations de la marchandise (chargement,déchargement, etc...) ou une impossibilité de passer par le noeud. Bien que les pénalités de transfert peuvent être utilisées dans d'autres buts, il serait bon d'en limiter l'utilisation uniquement à ce qui a été indiqué. Nous avons aussi des fonctions de coûts linéaires ou convexes à chaque lien pour chaque catégories de produits et des capacités modélisées par des fonctions linéaires par morceaux. Le réseau portuaire est un réseau multimode et multiproduit qui peut donc être intégré totalement dans un réseau global. Les flots d'exportation et d'importation sont représentés d'une façon symétrique. C'est la manière la plus naturelle de représenter le port puisque les ressources utilisées (entrepôts, élévateur de grains, ...) sont les mêmes pour les activités à l'exportation et à l'importation. La représentation proposée est très générale : chaque port possède sa propre représentation suivant les manipulations qui s'y déroulent et les produits qui y circulent.

Les composantes du réseau sont des noeuds. des centroïdes, des liens, des fonctions de transfert possiblement à chaque noeud, des fonctions de délai et des fonctions pour les capacités associées à chaque lien. Avec ces seules composantes, il est difficile de tenir compte de toutes les situations possibles. En gardant une représentation simple du réseau, on

perd la possibilité de modéliser toutes les situations pour considérer l'aspect multiproduit, multipériode et multiport. Dans le contexte du projet actuel sur la planification du transport brésilien, l'accent est mis sur l'exportation. La livraison et la réception de la marchandise sont plus planifiées à l'inportation qu'à l'exportation. Donc on peut considérer à l'importation que l'on a une demande d'un port déterminé du réseau à une certaine destination déterminée à l'intérieur du réseau de transport. Cela permet de résoudre les problèmes de détermination de la quantité de marchandise à entreposer à l'importation. A l'exportation, le modèle est assez souple pour pouvoir considérer des entreposages entre deux périodes d'analyse pour différents types de destination. En effet, le modèle permet d'avoir une destination dans le port, (modélisé par le centroïde A) ou une destination à l'extérieur du port (modélisé par le centroïde B) ou considérer une combinaison des deux. Nous déterminons au préalable la quantité maximale de produit pouvant être envoyée de chaque origine vers chaque destination en effectuant un calcul de flot maximal multiproduit. Nous pouvons alors séparer la demande à envoyer à la destination de la demande à entreposer.

La modélisation portuaire dépend de l'utilisation du modèle, de la gestion des marchandises et des politiques de planification. Il se pourrait que dans un autre contexte et dans d'autres situations, une autre modélisation soit plus adéquate. Dans le cadre du projet, les besoins d'analyse d'exportation et d'importation sont résolus et la modélisation est appropriée.

CONCLUSION

Dans le cadre d'une planification du transport de marchandises à l'échelle d'un pays, nous nous sommes attaqués aux deux aspects suivants : nous avons décrit un algorithme de calcul du flot maximal multiproduit pouvant être envoyé de chaque origine vers chaque destination dans un réseau à capacité limitée. Nous avons ensuite donné une méthodologie pour le traitement du réseau portuaire à l'intérieur de tout le réseau de transport de marchandises.

L'approche adoptée pour résoudre le problème du calcul du flot maximal multiproduit est l'utilisation de la méthode du **lagrangien augmenté**. L'aspect intéressant dans une approche duale telle le lagrangien augmenté est qu'en général, les contraintes de capacité ne sont pas des contraintes "dures", c'est-à-dire que dans la plupart des problèmes pratiques à résoudre, on permet que les contraintes soient légèrement "violées" (à approximativement 5 %). Considérons le sous-problème pour un multiplicateur lagrangien λ fixé. La résolution du sous-problème s'effectue par une approche Frank & Wolfe qui est une approche de réseau "pure", c'est-à-dire que l'on utilise les caractéristiques propres du réseau. D'autres approches peuvent être utilisées pour la résolution du sous-problème comme par exemple le simplexe convexe ou la décomposition Dantzig-Wolfe. Les avantages d'utiliser une approche de type Frank & Wolfe sont les suivants : les besoins en mémoire sont moins grands et la technique utilise la structure "pure" du réseau. Un désavantage se situe dans la convergence de la méthode : il est possible qu'une méthode telle le simplexe convexe converge plus rapidement que le lagrangien augmenté. Si l'on compare le problème du flot maximal multiproduit aux problèmes d'affectation de trafic, on peut noter que le problème traité équivaut

à résoudre plusieurs problèmes d'affectation de trafic à plusieurs produits. Notons également que la solution obtenue n'est pas unique et qu'elle est sensible aux paramètres suivants : les poids de chaque paire O/D i pour tout produit p γ_i^p, le facteur de pénalité PEN, la demande maximale pour chaque paire O/D i de chaque produit p $g_i^{p\ max}$ et les paramètres de précision des différents critères d'arrêt. Ces paramètres ainsi que tous les autres paramètres de précision ont été ajustés en développant un prototype. On a aussi montré que le modèle peut être étendu pour résoudre des problèmes plus complexes tel un modèle ajouté de contraintes pour borner inférieurement la quantité maximale de produit à envoyer. Les résultats obtenus ont montré que la performance de l'algorithme était satisfaisante au niveau de la qualité des solutions et du temps d'exécution.
Le calcul du flot maximal multiproduit est un calcul nécessaire dans l'étude d'un réseau avec capacité restreinte, comme c'est le cas dans les réseaux portuaires.

Trouver un modèle adéquat pour les ports est une opération complexe à cause des nombreuses activités qui se déroulent dans les terminaux. On doit tenir compte de nombreuses opérations et manipulations de la marchandise. Ce rapport décrit les différentes étapes qui ont mené à l'élaboration d'un modèle portuaire. Nous avons déterminé une représentation du port pouvant s'intégrer dans la représentation du reste du réseau de transport. Nous avons décrit les coûts et les fonctions de délai associées. Puisque le modèle général est statique, il a été nécessaire de définir une procédure générale pour l'entreposage des différents produits entre les périodes d'analyse aussi bien qu'à l'intérieur d'une même période. La difficulté dans la modélisation d'un port provient de la calibration des fonctions de coûts et de délais pour déterminer, par exemple, la quantité de marchandise à envoyer de la quantité à entreposer. L'aspect multiport ajoute à la complexité de la modélisation. De plus, l'aspect multiproduit du

réseau portuaire complique l'analyse puisque nous avons une fonction de délai pour chaque catégorie de produit. Finalement notons que le modèle présenté est un modèle général et que chaque port possède ses propres caractéristiques et donc sa propre représentation.

Suite à ce rapport, la prochaine étape consiste à expérimenter sur des réseaux tests le modèle portuaire ainsi qu'à intégrer dans l'outil de planification du transport de marchandises le modèle portuaire et le programme de calcul du flot maximal multiproduit. Cet aspect est actuellement en cours de développement par les équipes de P.U.C.-R.J., de GEIPOT et du Centre de Recherche sur les transports.

BIBLIOGRAPHIE

[1] **Assad, A.A.,**

"Multicommodity Network Flows- A survey Networks", Vol.8 (1978) p.37-91, John-Wiley & Sons, Inc.

[2] **Assad, A.A.,**

"Multicommodity Network Flows- Computational Experience", Working Paper OR-058-76, Oper. Res. Center, M.I.T., October 1976.

[3] **Babin, A.,**

"TRAFVAR for traffic assignments with variable demand", Publication #179, Centre de recherche sur les transports, Université de Montréal (septembre 1980).

[4] **Bertsekas,D.P.,**

"Constrained Optimization and Lagrange Multipliers Methods", Academic Press, New-York, 1982.

[5] **Boyer, D.D. & Hunt, D.J.,**

"A Modified Simplex Algorithm for Solving the Multicommodity Flow Problem", the George Washington University Logistics Research Project, Serial T-211, (September 1968).

[6] **Ford, L. R. & Fulkerson, D. R.,**

"A Suggested Computation for Maximal Multicommodity Network Flows" Management Science, Vol. 5, No.1 (1958), pp.97-101.

[7] **Gratzer, F. J. & Steiglitz, K.,**

"A Heuristic Approach to Large Multicommodity Flow Problems", Proc. of the Symposium on Computer Communications Networks and Teletraffic, ed. by J. Fox, 1972, pp. 311-324.

[8] **Grigoriadis, M.D. & White, W.W.,**

"A Partitionning Algorithm for the Multicommodity Network Flow Problem", Mathematical Programming, Vol. 3 , no 2, Octobre 1972, pp.157-177.

[9] **Grinold, R. C.,**

"A Multicommodity Max-Flow Algorithm", Operationnal Research, 16, 1968, pp. 1234-1238.

[10] **Guélat, J.,**

"Algorithmes pour le problème d'affectation du trafic d'équilibre avec demandes fixes - Comparaisons",
Publication #299, Centre de recherche sur les transports,
Université de Montréal (Mai 1983).

[11] **Hartman, J.K. & Lasdon, L.S.,**

"A Generalized Upper Bounding Algorithm for Multicommodity Network Flow Problems", Networks, (1972), pp.333-354.

[12] **Held, M., Wolfe, P. & Crowder, H. P.,**

"Validation of Subgradient Optimization",
Mathematical Programming, Vol. 6, 1, February 1974, pp. 62-88.

[13] **Jewell, W.S.,**

"A Primal-Dual Multicommodity Flow Algorithm", Orc 66-24,
Operations Research Center, University of California, Berkeley (1966).

[14] **Kennington, J. L. & Shalaby, M.,**

"An Effective Subgradient Procedure for Minimal Cost Multicommodity Flow Problems",
Tech. Report IEOR 750010, Dept. of Ind. Eng. & Oper. Res.,
Southern Methodist University, March 1976.

[15] **Luenberger,David G.,**

"Linear and Nonlinear Programming",
Stanford University, Addison-Wesley, second edition

[16] **Nguyen,S., & Dupuis,C.,**

"Une méthode efficace de calcul d'un trafic d'équilibre dans les cas des coûts non-symétriques"
Publication #205, Centre de Recherche sur les Transports,
Université de Montréal (1981).

[17] **Nguyen,S., & James,L.,**

"TRAFFIC, an equilibriun Traffic Assignment Program",
Publication #17, Centre de Recherche sur les Transports,
Université de Montréal (1975).

[18] **Robacker, J. T.,**

"Notes on Linear Programming : Part XXXVII Concerning Multicommodity Networks", RM-1799, The Rand Corporation, 1956.

[19] **Rothschild, B. & Whinston, A.,**

"On Two Commodity Network Flows",
Operationnal Research, 14, May-June 1966, pp.1121-1129.

[20] Saigal, R.,

"Multicommodity Flows in Directed Networks", ORC 69-14,
Operations Res. Center, University of California, Berkeley, September 1969.

[21] Sakarovitch, Michael,

"The Multicommodity Maximum Flow Problem", ORC 66-25,
Operations Research Center, University of California, Berkeley, 1966.

[22] Sakarovitch, Michael,

"The Multicommodity Max Flow Problem", ORC 69-14,
Operations Res. Center, University of California, Berkeley, September 1969.

[23] Sakarovitch, Michael,

"Two-Commodity Network Flows Linear Programming",
Mathematical Programming, 4, 1, February 1973, pp. 1-20.

[24] Sauvé, M.,

"Résolution Numérique du Problème de Multiflot Compatible à Coût
Minimum", Centre de Recherche sur lesTransports, Publication # 409,
Université de Montréal, Octobre 1984.

[25] Swoveland, C.,

"Decomposition Algorithms for the Multicommodity Distribution
Problem", Working Paper #184, Western Man. Science Inst.,
University of California, Los Angeles, October 1971.

[26] Wilness,P. & Franfel,E.,

"Port Analysis and Planning" ,
First International - Transportation research.

REMERCIEMENTS

J'aimerais remercier mon directeur de maîtrise Monsieur Michael Florian ainsi que Monsieur Teodor Crainic pour leur appui, leurs conseils et leur disponibilité tout au long de ce stage. Je tiens également à remercier tout le reste de l'équipe du projet 'Brésil' pour avoir aider à rendre le projet très intéressant.

Parmi l'équipe du Brésil, je tiens tout particulièrement à remercier Monsieur Luis Francisco Modenese Vieira de GEIPOT et Monsieur Jose Eugenio Leal de l'Université catholique de Rio de Janeiro pour l'intérêt qu'ils ont porté à mon stage et leurs remarques et conseils.

De plus, je désire adresser mes remerciements à Isabelle Constantin et à Diane Larin pour leur accessibilité et leur aide dans l'utilisation du PIXEL et du logiciel EMME/2.

ANNEXE A : Algorithme de la méthode PARTAN

Soit $f : E^n \dashrightarrow E^1$ une fonction différentiable et convexe.

Considérons le problème de minimisation à contraintes linéaires à résoudre :

$$\begin{aligned} &\text{Min } f(x) \\ &\text{sujet à : } Ax = b \\ &\qquad x \geq 0 \end{aligned}$$

L'algorithme de la méthode PARTAN s'énonce comme suit :

étape 0 (Initialisation)

Trouver une solution réalisable x^1 ; $r=1$ (indice d'itération)

étape 1 (Direction de descente Frank & Wolfe)

Résoudre le problème linéaire : $\text{Min } f(x^r) + \nabla f(x^r)(y-x^r)$ sujet à : $Ay = b$, $y \geq 0$

qui est équivalent au problème : $\text{Min } \nabla f(x^r)\, y$ sujet à : $Ay = b$, $y \geq 0$

Soit y^r une solution optimale du sous-problème.

(test d'optimalité) si $| (\nabla f(x^r)(y^r - x^r)) / f(x^r) | < \varepsilon$ alors STOP

La direction de descente est : $d^r = y^r - x^r$

étape 2 (Recherche unidimensionnelle dans la direction Frank & Wolfe)

Résoudre le problème $\text{Min } f(x^r + \delta d^r)$, $0 \leq \delta \leq 1$

Soit δ^r ("pas Frank & Wolfe") la solution obtenue.

$z^{r+1} = x^r + \delta^r d^r$

<u>étape 3</u> (Recherche unidimensionnelle dans la direction PARTAN)

Si r=1 alors $x^{r+1} = z^r$

Sinon *debut (du sinon)*

la direction de descente PARTAN est : $d_p^r = z^r - x^{r-1}$

On calcule le pas maximal p^{max} dans cette direction en effectuant une recherche unidimensionnelle.

On résoud : Min $f(x^{r-1} + p\, d_p^r)$

$0 \leq p \leq p^{max}$

Soit p^r ("pas PARTAN") la solution optimale de ce problème.

$x^{r+1} = x^{r-1} + p^r\, d_p^r$

fin (du sinon)

r = r+1

retour à l'étape 1.

ANNEXE B : "Listing" du programme "MAXFLO"

```
      PROGRAM MAXFLO(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3)
C
C     ----THIS PROGRAM DETERMINES THE MAXIMUM FLOW OF PRODUCTS THAT MAY BE
C          SENT FROM EACH ORIGIN TO EACH DESTINATION. THE METHOD USED TO
C          SOLVE THIS PROBLEM IS THE AUGMENTED LAGRANGIAN BASED ON :
C
C          "CONSTRAINED OPTIMIZATION AND LAGRANGE MULTIPLIERS METHODS "
C                          BY BERTSEKAS, D.P.
C          (ACADEMIC PRESS, NEW-YORK, 1982)
C
C     -----THIS PROGRAM WAS CODED BY NATHALIE RICO
C          AT THE TRANSPORTATION RESEARCH CENTER, UNIVERSITY OF MONTREAL (1985)
C
      COMMON /DIMENS/ NNODE,NLINK,NCENT,NPROD,NPAIR
      COMMON /NET1/ NOLAB(101),NDEST(1000),CAP(1000),FFLOW(1000),
     +              IPROD(1000)
      COMMON /NET2/ NPLAB(100),NPOD(350),GAMMA(5,350),BOUND(5,350)
     +              ,GMIN(5,350)
      COMMON /NET3/ FLOW(1000),FLOW2(1000),DIRF(1000),TRAV(1000),
     +  GPG(5,350),GPG2(5,350),DIRG(5,350)
      COMMON /PARAMG/EPSIG,MAXITG
      COMMON /UNIT/ NETFIL,RESFIL
      COMMON /DUAL/ U,OMEGA(1000)
      INTEGER NETFIL,RESFIL
C
 2000 FORMAT(//" ITERATION NUMBER ",I4///
     +          "      NUMBER OF ITERATIONS EXECUTED :  ",I4/
     +          "      VALUE OF OBJECTIVE FUNCTION   :  ",E25.15/
     +          "      VALUE OF THE GAP              :  ",E25.15/
     +          "      NORM OF THE DIFFERENCE BETWEEN"/
     +          "      MULTIPLIERS OF TWO  SUCCESSIVE"/
     +          "      ITERATIONS                    :  ",E25.15/)
 3000 FORMAT(/
     + 3X,"LINK  NODE  NODE    CAPACITY   FIXED      COST   PREDICTED  "
     +"MULTIPLIER   PRODUCTS ALLOWED"/3X," NB   FROM   TO",16X,"FLOW",
     + 15X,"FLOW"/)
 3010 FORMAT(2X,2(I4,1X),1X,I4,3X,3(F9.2,1X),F8.2,1X,F9.2,11X,I4)
 3020 FORMAT(1H1//
     + 3X,"O/D  NODE  NODE   WEIGHT     MIN      MAX     PREDICT   WEIGHT
     +  MIN      MAX     PREDICT   WEIGHT     MIN      MAX      PREDICT"/
     + 3X,"PAIR FROM   TO ",28X,"DEMAND",29X,"DEMAND",
     + 29X,"DEMAND"/)
 3030 FORMAT(2X,3(I4,1X),3(F8.2,1X,F8.2,1X,F8.2,1X,F8.2))
 3050 FORMAT(17X,3(F8.2,1X,F8.2,1X,F8.2,1X,F8.2)/)
 4000 FORMAT(1X,F15.2)
 4010 FORMAT(1X,5F15.2)
C
      NETFIL=1
      RESFIL=2
      REWIND RESFIL
C
C-----READ NETWORK DATA
C
      CALL NETWOR(U,IERR)
      IF(IERR.EQ.0) GOTO 1
      PRINT 1000,IERR
 1000 FORMAT(/5X,"ECO ERROR IN EXECUTION OF SUBROUTINE NETWOR :",I4)
C
C-----INITIALIZATIONS (STEP 0)
C
    1 ITER=0
      DO 5 J=1,NLINK
    5 OMEGA(J)=0.0
      DIFF=0.
C
C     ----SOLVE THE AUGMENTED LAGRANGIAN SUBPROBLEM (STEP 1)
C
   10 CALL TRAFFIC(ITER,ITERAT,OBF,GAP,DIFF,IERR)
      IF(IERR.EQ.0) GOTO 20
C
C     ----PRINT ERROR MESSAGE
C
      PRINT 1010,IERR
 1010 FORMAT(/5X,"E10 ERROR IN EXECUTION OF SUBROUTINE TRAFFIC :  ",I4)
      STOP 110
C
C-----ADJUSTMENT OF THE MULTIPLIERS (STEP 2)
C     COMPUTE STOPPING CRITERION # 1 :
C     NORM OF (OMEGA(ITER+1) - OMEGA(ITER))
```

```
20 DIFF=0.
   DO 30 J=1,NLINK
      FLOWI=0.
      IF(CAP(J).LT.0.) FLOWI=FLOW(JINV(J))
      VALMAX=U * (FLOW(J)+FLOWI+FFLOW(J)-ABS(CAP(J)))
      VALMAX2=-OMEGA(J)
      IF(VALMAX.LT.VALMAX2) VALMAX=VALMAX2
      DIFF=VALMAX**2 + DIFF
      VALMAX=OMEGA(J) + U * (FLOW(J)+FLOWI+FFLOW(J)-ABS(CAP(J)))
      IF(VALMAX.LT.0.0) VALMAX = 0.0
      OMEGA(J)=VALMAX
30 CONTINUE
   DIFF=SQRT(DIFF).
   ITER=ITER+1

----PENALTY FACTOR U CAN BE INCREMENTED AFTER A FEW ITERATIONS.
   THIS HELPS TO GET CLOSER TO THE SET OF FEASIBLE SOLUTIONS.

   IF(ITER.EQ.10) U=U+1

----COMPUTE STOPPING CRITERION # 2 : THE MAXIMUM OF THE DIFFERENCE
   BETWEEN THE PREDICT FLOW OF TWO SUCCESSIVE ITERATIONS.

----COMPUTE STOPPING CRITERION # 3 : THE MAXIMUM OF THE DIFFERENCE
   BETWEEN THE PREDICT DEMAND OF TWO SUCCESSIVE ITERATIONS.

   IF(ITER.NE.1) GOTO 60
   REWIND 3
   DO 40 I=1,NLINK
40 WRITE(3,4000) FLOW(I)
   DO 50 I=1,NPAIR
50 WRITE(3,4010) (GPQ(K,I),K=1,NPROD)

----STOPPING CRITERION FOR THE FIRST ITERATION

   IF(DIFF.LT.EPSIG.OR.ITER.GT.MAXITG) GOTO 110
----GO TO STEP 1
   GOTO 10
60 VMAXF=0.
   VMAXG=0.
   ILINK=1
   IPAIR=1
   KPROD=1
   REWIND 3
   DO 70 I=1,NLINK
      READ(3,4000) DIRF(I)
      VMAXF2=ABS(FLOW(I)-DIRF(I))
      IF(VMAXF2.LE.VMAXF) GOTO 70
      VMAXF=VMAXF2
      ILINK=I
70 CONTINUE
   DO 85 I=1,NPAIR
   READ(3,4010) (DIRG(K,I),K=1,NPROD)
   DO 80 K=1,NPROD
      VMAXG2=ABS(GPQ(K,I)-DIRG(K,I))
      IF(VMAXG2.LE.VMAXG) GOTO 80
      VMAXG=VMAXG2
      IPAIR=I
      KPROD=K
80 CONTINUE
85 CONTINUE
   REWIND 3
   DO 90 I=1,NLINK
90 WRITE(3,4000) FLOW(I)
   DO 100 I=1,NPAIR
00 WRITE(3,4010) (GPQ(K,I),K=1,NPROD)

 --STOPPING CRITERION

   IF(DIFF.LT.EPSIG.OR.VMAXF.LT.0.5*10E-4*ABS(CAP(ILINK)).OR.
  +VMAXG.LT.0.5*10E-4*DIRG(KPROD,IPAIR).OR.ITER.GT.MAXITG)GOTO 110
 --GO TO STEP 1
   GOTO 10

 --PRINT RESULTS : THE CALCULATED LINK FLOWS
                   THE O/D PAIR DEMANDS

 0 PRINT 2000,(ITER-1),ITERAT,OBF,GAP,DIFF
   PRINT 3000
   NO=1
```

```
      NO=1
      DO 120 I=1,NLINK
        NO=NORIG(NOLAB,I,NNODE)
        PRINT 3010,I,NO,NDEST(I),ABS(CAP(I)),FFLOW(I),TRAV(I),
     + FLOW(I),OMEGA(I),IPROD(I)
  120 CONTINUE
      PRINT 3020
      NO=1
      LIM=3
      IF(NPROD.LT.LIM) LIM=NPROD
      DO 130 K=1,NPAIR
        NO=NORIG(NPLAB,K,NCENT)
        PRINT 3030,K,NO,NPOD(K),(GAMMA(I,K),GMIN(I,K),BOUND(I,K)
     +  ,GPG(I,K),I=1,LIM)
        IF(NPROD.LE.3) GOTO 130
        PRINT 3050,(GAMMA(I,K),GMIN(I,K),BOUND(I,K),GPG(I,K),
     +  I=4,NPROD)
  130 CONTINUE
      STOP
      END

*****************************************************************

      SUBROUTINE NETWOR(U,IERR)

 ----READ IN NETWORK DATA FOR ALL LINK AND FOR ALL O/D PAIR

      COMMON /DIMENS/ NNODE,NLINK,NCENT,NPROD,NPAIR
      COMMON /NET1/ NOLAB(101),NDEST(1000),CAP(1000),FFLOW(1000),
     +              IPROD(1000)
      COMMON /NET2/ NPLAB(100),NPOD(350),GAMMA(5,350),BOUND(5,350)
     +              ,GMIN(5,350)
      COMMON /PARAMG/EPSIG,MAXITG
      COMMON /PARAM/EPSI,MAXIT,ACCUR
      COMMON /UNIT/ NETFIL,RESFIL
      INTEGER NETFIL,RESFIL

      INTEGER PARM,SIZE
      DATA PARM/"PARM"/,SIZE/"SIZE"/

 1010 FORMAT(1X,A10)
 1011 FORMAT(1H1//"C",24X,A10//)
 1015 FORMAT(1X,A4,I4,F5.3,F5.3)
 1016 FORMAT(1X,"NUMBER MAXIMUM OF ITERATIONS FOR THE MAXFLO :  ",I4/
     +       " EPSILON FOR THE STOPPING CRITERION :  ",E25.15/
     +       " VALUE OF THE PENALTY TERM U        :  ",E25.15/)
 1020 FORMAT(1X,A4,I4,F8.5,F10.5)
 1025 FORMAT(1X,"NUMBER MAXIMUM OF ITERATIONS FOR THE F&W PROBLEM :",
     +       I4/" EPSILON FOR THE STOPPING CRITERION :  ",
     +    E25.15/" ACCURACY FOR THE LINE SEARCH      :  ",
     +    E25.15/)
 1030 FORMAT(1X,A4,4I6)
 1035 FORMAT(/" NUMBER OF NODES      :  ",I4
     +       /" NUMBER OF LINKS      :  ",I4
     +       /" NUMBER OF CENTROIDS  :  ",I4
     +       /" NUMBER OF PRODUCTS   :  ",I4/)
 1040 FORMAT(1X,2(I5,1X),2F10.5,I4)
 1050 FORMAT(1X,2(I5,1X),15(F7.2))
 2010 FORMAT (/5X,"E02 LINKS OR DEMANDS NOT SORTED. LINE :",2I5)
 2020 FORMAT (/5X,"E01 NO. CARDS .NE. NLINKS ",2I8)
 2040 FORMAT (/5X,"E04 PARAMETER CARD MISSING :  ",A5)
 2050 FORMAT (/5X,"E05 SIZE CARD MISSING :  ",A5)
 2060 FORMAT (/5X,"E06 OVERFLOW IN INPUT DATA DIMENSIONS :",4I5)
      IERR=0
      ICARD=0

-----READ PROJECT TITLE

      READ 1010,TITLE
      PRINT 1011,TITLE

-----READ GLOBAL PARAMETERS FOR THE MAX-FLOW PROCEDURE

      READ 1015,ID,MAXITG,EPSIG,U
      PRINT 1016,MAXITG,EPSIG,U
      IF(ID.EQ.PARM) GOTO 10
         PRINT 2040,ID
         IERR=4
         RETURN
```

```
C
C ----READ ALGORITHM CONTROL PARAMETERS FOR THE FRANK & WOLFE SUBPROBLEM
C
   10 READ  1020, ID, MAXIT, EPSI, ACCUR
      PRINT 1025, MAXIT, EPSI, ACCUR
      IF (ID .EQ. PARM) GOTO 20
           PRINT 2040, ID
           IERR = 4
           RETURN
   20 IF(EPSI.LE.0.0) EPSI=0.05
      IF(MAXIT.LE.0) MAXIT=15
   30 READ 1030, ID, NNODE, NLINK, NCENT, NPROD
      PRINT 1035, NNODE, NLINK, NCENT, NPROD
      IF (ID .EQ. SIZE) GOTO 40
           PRINT 2050, ID
           IERR = 5
           RETURN
C
C-----CHECK INPUT DATA DIMENSIONS
C
   40 IF(NNODE.GT.100) GOTO 50
      IF(NLINK.GT.1000) GOTO 50
      IF(NCENT.GT.100) GOTO 50
      IF(NPROD.GT.5) GOTO 50
      GOTO 55
   50 IERR=6
      PRINT 2060, NNODE, NLINK, NCENT, NPROD
      RETURN
C
C-----READ NETWORK DATA
C
C LINK--1=NODE ORIGIN, 2=NODE DESTINATION, 3=CAPACITY, 4=FIXED FLOW
C       5=PRODUCTS ALLOWED ON A LINK
C
   55 DO 60 I=1,NNODE
   60    NOLAB(I)=0
      NOLAB(1)=1
      NOLAB(NNODE+1) = NLINK + 1
      I=0
      J=1
   70 I=I+1
      IF(I.GT.NLINK) GOTO 120
      READ(NETFIL,1040) NO,NDEST(I),CAP(I),FFLOW(I),IPROD(I)
C
C-----SET UP LINK ORIGIN INDEX TABLE NOLAB
C
      IF(NO-J)90,70,100
   90      IERR=1
           PRINT 2010, NO, J
           RETURN
  100 J=NO
      NOLAB(J)=I
      GOTO 70
C
C-----READ IN THE DEMANDS ASSOCIATED WITH THE CURRENT ORIGIN
C LINK--1=CENTROID ORIGIN, 2=CENTROID DESTINATION, 3=WEIGHT FOR EACH PRODUCT
C       4=MAXIMUM DEMAND FOR EACH PRODUCT
C
  120 DO 130 I=1,NCENT
  130 NPLAB(I)=0
      NPLAB(1)=1
      NPAIR=0
      I=0
      J=1
  140 I=I+1
      READ (NETFIL,1050) NO,NPOD(I),(GAMMA(K,I),K=1,NPROD),
     +   (BOUND(K,I),K=1,NPROD),(GMIN(K,I),K=1,NPROD)
      IF(EOF(NETFIL).NE.0) GOTO 170
C
C-----SET UP DEMAND ORIGIN INDEX TABLE NPLAB
C
      NPAIR=NPAIR+1
      IF(NO-J) 150,140,160
  150   IERR=1
        PRINT 2010, NO, J
        RETURN
  160 J=NO
      NPLAB(J)=I
      GOTO 140
  170 NPLAB(NCENT+1)=NPAIR+1
      RETURN
```

```
C*************************************************************************
C
      SUBROUTINE TRAFFIC(ITER, ITERAT, OBF, GAP, DIFF, IERR)
C
CCCCC-----THIS SUBROUTINE COMPUTES "WARDROP" EQUILIBRIUM  FLOWS  WITH
CCCCC     VARIABLE DEMAND FUNCTION ON A NETWORK SUBJECT TO CONGESTION.
CCCCC
CCCCC-----THE SUBROUTINE IS BASED ON :
CCCCC     1)    "A MATHEMATICAL PROGRAMMING APPROACH TO EQUILIBRIUM
CCCCC     OF TRAFFIC ASSIGNMENT WITH FIXED DEMANDS" BY SANG NGUYEN
CCCCC     (UNIVERSITE DE MONTREAL, DEPARTEMENT D' INFORMATIQUE,
CCCCC     TECHNICAL REPORT,  NUMBER 138,  AUGUST 1973)
CCCCC
CCCCC     MODIFIED TO INCLUDE PARTAN TECHNIQUE BY J. GUELAT.
CCCCC
CCCCC     2)    SUBROUTINE "TRAFVAR" FOR TRAFFIC ASSIGNMENTS WITH VARIABLE
CCCCC     DEMAND BY ANDRE BABIN
CCCCC     (UNIVERSITE DE MONTREAL,  CENTRE DE RECHERCHE SUR LES TRANSPORTS
CCCCC     PUBLICATION # 179, SEPTEMBRE 1980)
C
      COMMON /DIMENS/ NNODE, NLINK, NCENT, NPROD, NPAIR
      COMMON /NET1/ NOLAB(101), NDEST(1000), CAP(1000), FFLOW(1000),
     +              IPROD(1000)
      COMMON /NET2/ NPLAB(100), NPOD(350), GAMMA(5, 350), BOUND(5, 350)
     +              , GMIN(5, 350)
      COMMON /NET3/ FLOW(1000), FLOW2(1000), DIRF(1000), TRAV(1000),
     +  GPG(5, 350), GPG2(5, 350), DIRG(5, 350)
      COMMON /PARAM/EPSI, MAXIT, ACCUR
      COMMON /UNIT/ NETFIL, RESFIL
      COMMON /DUAL/ U, OMEGA(1000)
      DIMENSION FLOW1(1000), GPG1(5, 350), WPG(1001), LARC(1001)
      INTEGER NETFIL, RESFIL
      REAL LAMBDA, LAMBK1
C
CCC
 1000 FORMAT(//" ITERATION NUMBER ", I4///
     +           "      NUMBER OF ITERATIONS EXECUTED :  ", I4/
     +           "      VALUE OF OBJECTIVE FUNCTION   :  ", E25.15/
     +           "      VALUE OF THE GAP              :  ", E25.15/
     +           "      NORM OF THE DIFFERENCE BETWEEN"/
     +           "      MULTIPLIERS OF TWO  SUCCESSIVE"/
     +           "      ITERATIONS                    :  ", E25.15/)
 2010 FORMAT(1H1, 30X, 5A8//)
 2005 FORMAT(I4, 3(I4, 4X), F6.2, 4X, I2, 6X, F6.2, F12.4)
 3000 FORMAT(/
     + 3X, "LINK  NODE  NODE    CAPACITY   FIXED      COST   PREDICTED  "
     +"MULTIPLIER   PRODUCTS ALLOWED"/3X," NB   FROM   TO", 16X, "FLOW",
     + 15X, "FLOW"/)
 3010 FORMAT(2X, 2(I4, 1X), 1X, I4, 3X, 3(F9.2, 1X), F8.2, 1X, F9.2, 11X, I4)
 3020 FORMAT(//
     + 3X, "O/D  NODE  NODE   WEIGHT    MIN      MAX     PREDICT   WEIGHT
     +  MIN      MAX     PREDICT   WEIGHT     MIN      MAX      PREDICT"/
     + 3X, "PAIR FROM   TO ", 28X, "DEMAND", 29X, "DEMAND",
     + 29X, "DEMAND"/)
 3030 FORMAT(2X, 3(I4, 1X), 3(F8.2, 1X, F8.2, 1X, F8.2, 1X, F8.2))
 3050 FORMAT(17X, 3(F8.2, 1X, F8.2, 1X, F8.2, 1X, F8.2)/)
 4000 FORMAT(/5X, "E11 THERE IS AN O/D PAIR WHERE THE IMPEDANCE IS ZERO",
     +  I4)
 4010 FORMAT(/5X, "E12 NO PATH IN NETWORK FOR O/D PAIR", I4, 1X, I4,
     +  " OF PRODUCT ", I1, " ERROR NUMBER ", I4)
C
CC-----INITIALIZATIONS
C
      ITERAT = 0
      IERR=0
      OBF=1
C
CCCCC*************************************** BEGINNING OF THE ALGORITHM
CCCCC
CCCCC     I----------I
CCCCC     I  STEP 0  I
CCCCC     I----------I
CCCCC
C
C-----SET UP INITIAL (ALL OR NOTHING) SOLUTION
C
      DO 10 I=1, NLINK
   10    FLOW(I)=0.0
      DO 15 I=1, NLINK
   15    TRAV(I)=USERCST(I)
```

```
          GPG(I,J)=0.
   20 CONTINUE
   30 CONTINUE
C
C
C----FIND BOUNDS FOR THE SUB-PROBLEM OF FRANK-WOLFE
C
      DO 70 I=1,NCENT
      IF(NPLAB(I).LE.0) GOTO 70
C
C-----CONSTRUCT SHORTEST PATHS FROM ORIGIN I TO ALL DESTINATIONS
C     FOR EACH PRODUCT
C
      DO 68 K=1,NPROD
      CALL MINTRE(I,WPG,LARC,K)
      II=I
      I1=NPLAB(II)
   40 II=II+1
      I2=NPLAB(II)-1
      IF(I2.LE.0)GOTO 40
      DO 65 L=I1,I2
      JK=NPOD(L)
      WVAL=WPG(JK)
      IF(ABS(WVAL).GE.0.0) GOTO 50
      IERR=3
      PRINT 4000,IERR
      RETURN
   50 GVAL=GMIN(K,L)
      GPG(K,L)=GMIN(K,L)
      NEXT=NPOD(L)
C
C-----TRACE BACK LINKS ON SHORTEST PATH FOR EACH DESTINATION AND
C     ADD OD VOLUME TO LINK TOTAL.
C
   60 LL=LARC(NEXT)
      IF(LL.EQ.0)GOTO 80
      FLOW(LL)=FLOW(LL)+GVAL
      NEXT=NORIG(NOLAB,LL,NNODE)
      IF(NEXT.EQ.I)GOTO 65
      GOTO 60
   65 CONTINUE
   68 CONTINUE
   70 CONTINUE
      GOTO 100
C
C----PRINT ERROR MESSAGE IF THERE IS NO PATH
C
   80 IERR=5
      PRINT 4010,I,JK,K,IERR
      RETURN
C
C
C     I----------I
C     I  STEP 1  I
C     I----------I
C
C
C
C-----INITIALIZATIONS
C
  100 DO 110 I=1,NLINK
        DIRF(I)=0.
        FLOW2(I)=FLOW(I)
        TRAV(I)=USERCST(I)
  110 CONTINUE
      DO 120 I=1,NPROD
      DO 120 J=1,NPAIR
        DIRG(I,J)=0
        GPG2(I,J)=GPG(I,J)
  120 CONTINUE
      HDEMG=0
      DO 185 I=1,NCENT
      IF(NPLAB(I).LE.0)GOTO 185
C
C-----CONSTRUCT SHORTEST PATHS FROM ORIGIN I TO ALL DESTINATIONS
C     FOR EACH PRODUCT
C
      DO 182 K=1,NPROD
      CALL MINTRE(I,WPG,LARC,K)
```

```
      II=1
      I1=NPLAB(II)
  130 II=II+1
      I2=NPLAB(II)-1
      IF(I2.LE.0)GOTO 130
      DO 180 L=I1,I2
      JK=NPOD(L)
      WVAL=WPG(JK)
  140 DIF=ABS(GAMMA(K,L)-WVAL)/GAMMA(K,L)
      VAL=0.
      IF(DIF.GT.0.01) GOTO 150
      DIRG(K,L)=GPG(K,L)
      GOTO 160
  150 IF(GAMMA(K,L).LT.WVAL)GOTO 155
      DIRG(K,L)=BOUND(K,L)
      GOTO 160
  155 DIRG(K,L)=GMIN(K,L)
  160 VAL=DIRG(K,L)
      NEXT=NPOD(L)
C
CC ----TRACE BACK LINKS ON SHORTEST PATH FOR EACH DESTINATION AND
CC     ADD OD VOLUME TO LINK TOTAL.
C
  165 LL=LARC(NEXT)
      IF(LL.EQ.0)GOTO 190
      DIRF(LL)=DIRF(LL)+VAL
      NEXT=NORIG(NOLAB,LL,NNODE)
      IF(NEXT.EQ.I) GOTO 170
      GOTO 165
  170 HDEMG=HDEMG+GAMMA(K,L)*(DIRG(K,L)-GPG(K,L))
  180 CONTINUE
  182 CONTINUE
  185 CONTINUE
      GOTO 192
C
CC ----PRINT ERROR MESSAGE IF THERE IS NO PATH
C
  190 IERR=5
      PRINT 4010,I,JK,K,IERR
      RETURN
  192 HDEMF=0.
      DO 195 I=1,NLINK
        DIRF(I)=DIRF(I)-FLOW(I)
        HDEMF=HDEMF+TRAV(I)*DIRF(I)
  195 CONTINUE
      DO 197 I=1,NPROD
      DO 196 J=1,NPAIR
        DIRG(I,J)=DIRG(I,J)-GPG(I,J)
  196 CONTINUE
  197 CONTINUE
      ITERAT=ITERAT+1
C
CC ----CHECK THE NUMBER OF ALLOWABLE ITERATIONS
C
      IF (ITERAT.LT.MAXIT) GOTO 198
      GOTO 400
C
CC-----STOPPING CRITERION
C
  198 GAP=ABS(HDEMF-HDEMG)
      IF(ABS(OBF).LE.10E-10) GOTO 199
      IF(GAP/ABS(OBF).LE.EPSI) GOTO 400
      GOTO 200
  199 IF(GAP.LE.EPSI) GOTO 400
C
CCCCCC
CCCCCC     I----------I
CCCCCC     I  STEP 2  I      *** FRANK & WOLFE LINE SEARCH ***
CCCCCC     I----------I
CCCCCC
C
  200 CALL LS(LAMBDA,1.00,ACCUR,OBF,0)
C
C     WRITE(PASFIL,202) LAMBDA
  202 FORMAT(/21X,"PAS FRANK & WOLFE :  ",E15.5/)
C
CC ----NEW CURRENT SOLUTION
C
```

```
      DO 205 J=1,NLINK
  205   FLOW(J)=FLOW(J)+LAMBDA*DIRF(J)
      DO 210 J=1,NLINK
  210   TRAV(J)=USERCST(J)
      DO 230 I=1,NPROD
      DO 220 J=1,NPAIR
        GPQ(I,J)=GPQ(I,J)+LAMBDA*DIRG(I,J)
  220 CONTINUE
  230 CONTINUE
C
      IF (ITERAT.GE.2) GOTO 300
      LAMBK1=LAMBDA
      RHOK=0.0
      DO 250 J=1,NLINK
  250 FLOW1(J) = FLOW(J)
      DO 260 I=1,NPROD
      DO 260 J=1,NPAIR
  260   GPQ1(I,J)=GPQ(I,J)
      GOTO 100
C ----TO STEP 1
C
C
C     I----------I
C     I  STEP 3  I      *** PARTAN LINE SEARCH ***
C     I----------I
C
C
C-----SEARCH FOR THE MAXIMUM PARTAN STEP
C
  300 PMAX = 1.0E10
      AUX=(1.0-LAMBDA)*(1.0-RHOK)*(1.0-LAMBK1)
      LAMBK1=LAMBDA
      IF (AUX.GE.1.0) GOTO 305
        PMAX=1.0/(1.0-AUX)
  305 CONTINUE
C
C     WRITE(PASFIL,310) PMAX
  310 FORMAT(/21X,"PAS PARTAN MAX :  ",E15.5)
C
C-----PARTAN DIRECTION SEARCH
C
      DO 342 J=1,NLINK
  342 DIRF(J)=FLOW(J)-FLOW2(J)
      DO 345 I=1,NPROD
      DO 345 J=1,NPAIR
  345   DIRG(I,J)=GPQ(I,J)-GPQ2(I,J)
C
C-----PARTAN LINE SEARCH
C
      CALL LS (PAS,PMAX,ACCUR,OBF,1)
      IF (PAS.LE.1.0) PAS = 1.0
      RHOK = 0.0
      IF (PMAX.EQ.1.0) GOTO 340
        RHOK = (PAS-1.0)/(PMAX-1.0)
  340 CONTINUE
C
C
C     WRITE(PASFIL,350) PAS
  350 FORMAT(/21X,"PAS PARTAN :  ",E15.5)
C
C-----NEW CURRENT SOLUTION
C
      DO 360 J=1,NLINK
         FLOW(J) = FLOW2(J) + PAS*DIRF(J)
         FLOW2(J) = FLOW1(J)
         FLOW1(J) = FLOW(J)
  360 CONTINUE
      DO 365 J=1,NLINK
  365    TRAV(J) = USERCST(J)
      DO 380 I=1,NPROD
      DO 380 K=1,NPAIR
         GPQ(I,K) = GPQ2(I,K)+PAS*DIRG(I,K)
         GPQ2(I,K)=GPQ1(I,K)
         GPQ1(I,K)=GPQ(I,K)
  380 CONTINUE
      GOTO 100
C     TO STEP 1
C
C
C*************************************** END OF THE ALGORITHM
C
  400 CONTINUE
```

```
C----PRINT OUT IN FILE RESFIL    THE CALCULATED LINK FLOWS
C                                THE O/D PAIR DEMANDS
C
      WRITE(RESFIL,1000) ITER,ITERAT,OBF,GAP,DIFF
      WRITE(RESFIL,3000)
      NO=1
      DO 420 I=1,NLINK
        NO=NORIG(NOLAB,I,NNODE)
        WRITE(RESFIL,3010) I,NO,NDEST(I),ABS(CAP(I)),FFLOW(I),
     + TRAV(I),FLOW(I),OMEGA(I),IPROD(I)
  420 CONTINUE
      WRITE(RESFIL,3020)
      NO=1
      LIM=3
      IF(NPROD.LT.LIM) LIM=NPROD
      DO 430 K=1,NPAIR
        NO=NORIG(NPLAB,K,NCENT)
        WRITE(RESFIL,3030) K,NO,NPOD(K),(GAMMA(I,K),GMIN(I,K),
     +  BOUND(I,K),GPG(I,K),I=1,LIM)
        IF(NPROD.LE.3) GOTO 430
        WRITE(RESFIL,3050)(GAMMA(I,K),GMIN(I,K),BOUND(I,K),GPG(I,K),
     +  I=4,NPROD)
  430 CONTINUE
      RETURN
      END
C
C*****************************************************************************
C
      INTEGER FUNCTION NORIG(NLAB,LL,NNODE)
C
C ----THIS FUNCTION DETERMINES THE NODE ORIGIN
C
      DIMENSION NLAB(1001)
      NORIG=0
      NODTOT=NNODE+1
      DO 10 I=1,NODTOT
      IF(NLAB(I).LE.0)GOTO 10
      IF(NLAB(I).LE.LL) GOTO 15
      NORIG=J
      RETURN
   15 J=I
   10 CONTINUE
      RETURN
      END
C
C*****************************************************************************
C
      INTEGER FUNCTION JINV(J)
C
C-----LET J BE THE LINK NUMBER FROM NODE A TO NODE B
C     THEN THIS FUNCTION DETERMINES LINK NUMBER JINV
C     FROM NODE B TO NODE A
C
      COMMON /DIMENS/ NNODE,NLINK,NCENT,NPROD,NPAIR
      COMMON /NET1/ NOLAB(101),NDEST(1000),CAP(1000),FFLOW(1000),
     +              IPROD(1000)
      NO=NORIG(NOLAB,J,NNODE)
      J2=NOLAB(NDEST(J))
   10 IF(J2.EQ.NOLAB(NDEST(J)+1)) RETURN
      IF(NO.EQ.NDEST(J2)) GOTO 20
      J2=J2+1
      GOTO 10
   20 JINV=J2
      RETURN
      END
C
C*****************************************************************************
C
      INTEGER FUNCTION IPERM(IPRD,IP)
C
C-----THIS FUNCTION DETERMINES IF PRODUCT IP IS ALLOWED ON A CERTAIN LINK
C     IPERM=0 IF THE PRODUCT IP IS NOT ALLOWED ON THE LINK
C          =1 IF THE PRODUCT IP IS ALLOWED ON THE LINK
C
C     THIS FUNCTION IS CALLED BY SUBROUTINE "MINTRE"
C
      IF(IPRD.EQ.0) GOTO 20
      IF(IPRD.LT.-10.OR.IPRD.GT.10) GOTO 10
      IF(IPRD.GT.0.AND.IPRD.EQ.IP) GOTO 20
      IF(IPRD.LT.0.AND.IABS(IPRD).NE.IP) GOTO 20
      IPERM=0
```

```
      IPERM=0
      RETURN
   10 IPRD2=IABS(IPRD/10)
      IPRD3=IABS(IPRD-IPRD/10*10)
      IF(IPRD.GT.10.AND.(IPRD2.EQ.IP.OR.IPRD3.EQ.IP)) GOTO 20
      IF(IPRD.LT.-10.AND.IPRD2.NE.IP.AND.IPRD3.NE.IP) GOTO 20
      IPERM=0
      RETURN
   20 IPERM=1
      RETURN
      END
C
C*****************************************************************************
C
      SUBROUTINE  MINTRE (NN,TSUM,LPR,IP)
C
      COMMON /DIMENS/ NNODE,NLINK,NCENT,NPROD,NPAIR
      COMMON /NET1/ NOLAB(101),NDEST(1000),CAP(1000),FFLOW(1000),
     +              IPROD(1000)
      COMMON /NET2/ NPLAB(100),NPOD(350),GAMMA(5,350),BOUND(5,350)
     +              ,GMIN(5,350)
      COMMON /NET3/ FLOW(1000),FLOW2(1000),DIRF(1000),TRAV(1000),
     +  GPQ(5,350),GPQ2(5,350),DIRG(5,350)
      DIMENSION LPR(1001),TSUM(1001),NSET(1001)
C
C
C-----MIN PATH USING ESOPO-PAPE ALGORITHM
C
C     CONSTRUCT SHORTEST PATH TO ALL DESTINATIONS FROM
C     ORIGIN NN.
C     LPR(N)    WILL CONTAIN THE ENTRY LINK INCIDENT TO
C               NODE N.
C     TSUM(N)   WILL RECORD THE SHORTEST DISTANCE TO NODE N.
C     NSET(N)   IS THE LIST OF CANDIDATES FOR SCANNING
C           =0   IF N HAS NEVER BEEN SCANNED
C           =-1  IF N HAS BEEN SCANNED ONCE
C           =MAX   IF N IS THE LAST NODE TO BE SCANNED OR
C                  N IS A DESTINATION
C           ELSE NSET(N)  POINTS TO THE NEXT NODE IN
C                         THE LIST.
C     LAST      INDICATES THE BOTTOM OF THE LIST.
C     LPR(N)    IS THE CORRESPONDING LINK TO NODE N IN THE
C               SHORTEST PATHS TREE FROM NN.
C
C-----INITIALIZATION
C
      MAX=999999
      DO 10 I=1,NNODE
        LPR(I)=0
        TSUM(I)= 1.0E10
        NSET(I)=0
        IF(NOLAB(I).EQ.0)NSET(I)=MAX
   10 CONTINUE
C
C-----NO CENTROID WILL BE SCANNED BUT THE ORIGIN
C
      DO 20 I=1,NCENT
   20 NSET(I)=MAX
C
C
      LAST=NN
      N=NN
      TSUM(N)=0.0
      GOTO 110
C
C-----FETCH NEW NODE FOR SCANNING.  IF LIST EMPTY STOP
C
  100 I=N
      N=NSET(N)
      IF(N.EQ.MAX)GOTO 200
      NSET(I)=-1
C
C-----SCAN NODE N
C
  110 I1=NOLAB(N)
      K=N
  120 K=K+1
      I2=NOLAB(K)-1
      IF(I2.LT.0)GOTO 120
```

```
      DO 150 I=I1,I2
      M=NDEST(I)
      VAL=TSUM(N)+TRAV(I)
C
C ----CHECK IF PRODUCT IP IS ALLOWED ON LINK I
C
      IF(VAL.GE.TSUM(M).OR.IPERM(IPROD(I),IP).EQ.0) GOTO 150
      TSUM(M)=VAL
      LPR(M)=I
C
C ----ADD M TO LIST IF FIRST TIME REACH  M  OR LIST HAS
C     ONLY ONE ELEMENT, ELSE INSERT M AT TOP LIST IF
C     M IS NOT ALREADY IN LIST
C
      IF(NSET(M))130,140,150
  130 IF(NSET(N).EQ.MAX)GOTO 140
      NSET(M)=NSET(N)
      NSET(N)=M
      GOTO 150
  140 NSET(M)=MAX
      NSET(LAST)=M
      LAST=M
  150 CONTINUE
C
      GOTO 100
  200 CONTINUE
      RETURN
      END
C
C*************************************************************************
C
      REAL FUNCTION SUMTRA (LAMBDA,MODE)
C
C ----CALCULATE OBJECTIVE FUNCTION (SUM TRAVEL TIME) FOR A
C     GIVEN LAMBDA.
C
      COMMON /DIMENS/ NNODE,NLINK,NCENT,NPROD,NPAIR
      COMMON /NET1/ NOLAB(101),NDEST(1000),CAP(1000),FFLOW(1000),
     +              IPROD(1000)
      COMMON /NET2/ NPLAB(100),NPOD(350),GAMMA(5,350),BOUND(5,350)
     +              ,GMIN(5,350)
      COMMON /NET3/ FLOW(1000),FLOW2(1000),DIRF(1000),TRAV(1000),
     +  GPQ(5,350),GPQ2(5,350),DIRG(5,350)
      COMMON /DUAL/ U,OMEGA(1000)
      REAL LAMBDA
C
C
      SUM1=0.0
      SUM2=0.0
      DO 10 J=1,NLINK
        VOL=FFLOW(J)+(1-MODE)*FLOW(J)+MODE*FLOW2(J)+LAMBDA*DIRF(J)
        FLOWI=0.
        IF(CAP(J).LT.0) FLOWI=FLOW(JINV(J))
        VALMAX=OMEGA(J) + U * (VOL+FLOWI-ABS(CAP(J)))
        IF(VALMAX.LT.0) VALMAX=0.0
        SUM1=SUM1+VALMAX**2-OMEGA(J)**2
   10 CONTINUE
      DO 30 I=1,NPROD
      DO 20 K=1,NPAIR
        T=((1-MODE)*GPQ(I,K)+MODE*GPQ2(I,K)+LAMBDA*DIRG(I,K))*GAMMA(I,K)
        SUM2=SUM2+T
   20 CONTINUE
   30 CONTINUE
      SUMTRA =SUM1/(2*U)-SUM2
      RETURN
      END
C
C*************************************************************************
C
      REAL FUNCTION FPRIM (LAMBDA,MODE)
C
C ----CALCULATE THE DERIVATIVE OF THE OBJECTIVE
C     FUNCTION FOR A GIVEN LAMBDA.
C
```

```
      COMMON /DIMENS/ NNODE,NLINK,NCENT,NPROD,NPAIR
      COMMON /NET1/ NOLAB(101),NDEST(1000),CAP(1000),FFLOW(1000),
     +              IPROD(1000)
      COMMON /NET2/ NPLAB(100),NPOD(350),GAMMA(5,350),BOUND(5,350)
     +              ,GMIN(5,350)
      COMMON /NET3/ FLOW(1000),FLOW2(1000),DIRF(1000),TRAV(1000),
     +  GPG(5,350),GPG2(5,350),DIRG(5,350)
      COMMON /DUAL/ U,OMEGA(1000)
      REAL LAMBDA
C
C
      SUM1 = 0.0
      SUM2 = 0.0
      DO 10 J=1,NLINK
         VOL=FFLOW(J)+(1-MODE)*FLOW(J) + MODE*FLOW2(J) +LAMBDA*DIRF(J)
         FLOWI=0.
         IF(CAP(J).LT.0) FLOWI=FLOW(JINV(J))
         VALMAX=OMEGA(J) + U * (VOL+FLOWI-ABS(CAP(J)))
         IF(VALMAX.LT.0) VALMAX=0.0
         SUM1=SUM1+VALMAX*DIRF(J)
   10 CONTINUE
      DO 30 I=1,NPROD
      DO 20 K=1,NPAIR
         SUM2=SUM2+DIRG(I,K)*GAMMA(I,K)
   20 CONTINUE
   30 CONTINUE
      FPRIM = SUM1-SUM2
      RETURN
      END
C
C*************************************************************************
C
      REAL FUNCTION USERCST( J )
C
C-----CALCULATE USER COST = TRAVEL TIME FOR LINK J
C
      COMMON /DIMENS/ NNODE,NLINK,NCENT,NPROD,NPAIR
      COMMON /NET1/ NOLAB(101),NDEST(1000),CAP(1000),FFLOW(1000),
     +              IPROD(1000)
      COMMON /NET2/ NPLAB(100),NPOD(350),GAMMA(5,350),BOUND(5,350)
     +              ,GMIN(5,350)
      COMMON /NET3/ FLOW(1000),FLOW2(1000),DIRF(1000),TRAV(1000),
     +  GPG(5,350),GPG2(5,350),DIRG(5,350)
      COMMON /DUAL/ U,OMEGA(1000)
C
      FLOWI=0.
      IF(CAP(J).LT.0.) FLOWI=FLOW(JINV(J))
      VALMAX=OMEGA(J) + U * (FLOW(J)+FLOWI+FFLOW(J)-ABS(CAP(J)))
      IF(VALMAX.LT.0) VALMAX=0
      USERCST = VALMAX
      RETURN
      END
C
C*************************************************************************
C
      SUBROUTINE LS(LAMBDA,DELTA,ACCUR,VAL,MODE)
C
C-----FIND LAMBDA IN THE INTERVAL ZERO TO "DELTA" SUCH THAT
C     THE DERIVATIVE OF THE OBJECTIVE FUNCTION IS ZERO.
C     VAL IS SET TO THE CORRESPONDING OBJECTIVE FUNCTION
C     VALUE.
C
      REAL LAMBDA
C
      XA = 0.0
      XB = DELTA
      FPXA = FPRIM(XA,MODE)
      FPXB = FPRIM(XB,MODE)
C
      IF (FPXA*FPXB .LT. 0.0) GOTO 30
         FXA = SUMTRA(XA,MODE)
         FXB = SUMTRA(XB,MODE)
         IF (FXA-FXB) 10,20,20
   10       VAL = FXA
            LAMBDA = XA
            RETURN
   20       VAL = FXB
            LAMBDA = XB
            RETURN
```

```
C
C-----BISSECTION
C
      XMID = 0.5*(XA+XB)
      FPXMID = FPRIM(XMID,MODE)
      IF (FPXMID*FPXA .LT. 0.0) GOTO 40
         XA = XMID
         FPXA = FPXMID
         GOTO 50
   40    XB = XMID
         FPXB = FPXMID
   50 CONTINUE
C
C-----REGULA FALSI
C
      XSUIV = XB - ((XB-XA)/(FPXB-FPXA))*FPXB
      FPXSUI = FPRIM(XSUIV,MODE)
      IF (ABS(FPXSUI).GT.ACCUR) GOTO 60
         FXSUIV = SUMTRA(XSUIV,MODE)
         VAL = FXSUIV
         LAMBDA = XSUIV
         RETURN
C
   60 IF (FPXSUI*FPXA .LT. 0.0) GOTO 70
         XA = XSUIV
         FPXA = FPXSUI
         GOTO 30
   70    XB = XSUIV
         FPXB = FPXSUI
         GOTO 30
      END
```